গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

শূন্য নম্বর দিনে আমরা শূন্য থেকে একটা পিসিতে পৌছলাম। এক নম্বর দিনে আমরা সিপিইউ আর মেমরির একদম গোড়ার কিছু কথা বলে নিয়ে, একটা পিসির কাজ করে চলার মধ্যে হার্ডওয়ার, সফটওয়ার, অপারেটিং সিস্টেম ঠিক কী ভাবে সাজানো থাকে সেই নিয়ে আলোচনা করলাম। দুই নম্বর দিনে, ওই কাজ করে চলতে গিয়ে যে আইও বা ইনপুট আউটপুট ডিভাইসগুলোকে ব্যবহার করতে হয় তাদের সঙ্গে অপারেটিং সিস্টেমের দেওয়া-নেওয়া নিয়ে আলোচনা করলাম। তারপর, তিন নম্বর দিনে আমরা পিসির এই অবস্থায় পৌছনোর পিছনের ধারাবাহিকতাটা নিয়ে কথা বললাম — যন্ত্র এবং মনন দুই দিক দিয়ে। আমরা পিসির প্রাগ-ইতিহাস থেকে ইতিহাসে পৌঁছলাম। আজ আমরা পিসির ইতিহাস নিয়ে কথা বলব। পিসিতে আমরা পৌছলাম কোন বিবর্তনের মধ্যে দিয়ে, যন্ত্রের বিবর্তন, যন্ত্রকে ব্যবহার করার তরিকার পরিবর্তন, যন্ত্রের সঙ্গে যন্ত্রব্যবহারকারীর সম্পর্কের পরিবর্তন। যে অপারেটিং সিস্টেম নিয়ে দু-চারটে গোড়ার কথা আমরা এক আর দুই নম্বর দিনে বলে এসেছি, সেই অপারেটিং সিস্টেমটা গজিয়ে উঠল কী করে সেটা নিয়েই আজ কথা বলব আমরা। অপারেটিং সিস্টেম ঠিক কী এবং কী ভাবে কাজ করে, অপারেটিং সিস্টেমের শ্রেণীবিভাগ, ইত্যাদি। আমরা শুরু করব একদম আধুনিক ডিজিটাল কম্পিউটারের জন্মমুহূর্ত থেকে। প্রথমে সেই জন্মের অজস্র নাম-জানা এবং নাম-না জানা ধাত্রীদের ভিতর থেকে কয়েকজনকে নিয়ে একটু আলাদা করে কথা বলে নেব। তারপর এই প্রথম ছানা মানে জেনারেশনটা কী ভাবে পরবর্তী প্রজন্মে বেড়ে উঠতে শুরু করল তার কাহিনী। এটাও আবার এক অর্থে, আমাদের এই গ্নু-লিনাক্স ইশকুলের পাঠমালার সাপেক্ষে একটা প্রাককথন, যা শেষ হবে আজ। পরের দিন, মানে পাঁচে আমরা সরাসরি গ্নু-লিনাক্সে ল্যান্ড করে যাব। আজকের একদম শেষদিকে গিয়ে আমরা গ্নু-লিনাক্সের শুরুর শুরু সম্পর্কে কিছু কথা জানব। অবশিষ্টটা পরের দিন।

।। দিন চার ।।

১ ।। শুরুর কয়েকজন মানুষ এবং মেশিন

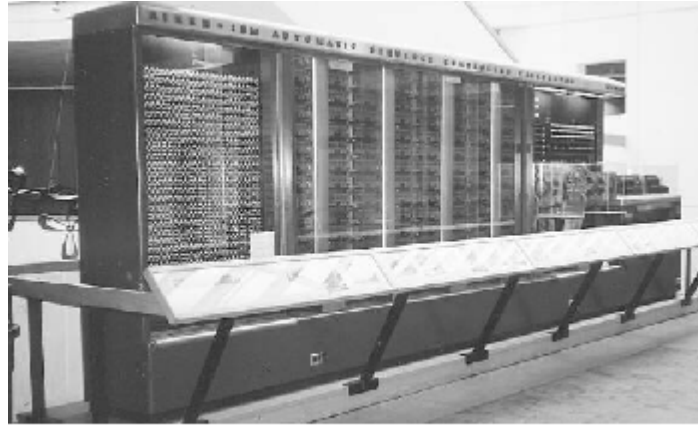
তিন নম্বর দিনে আমরা শ্যাননের লজিক সার্কিটের তত্ত্বের জন্মমুহূর্তে শেষ করেছিলাম। ওখান থেকে আধুনিক ডিজিটাল কম্পিউটারের প্রাগ-ইতিহাসের শেষ এবং ইতিহাসের শুরু। ইতিহাসে এর পরের স্তরেই আসেন ১৯৩০ নাগাদ হাওয়ার্ড আইকেন তার মার্ক ওয়ান নামে মেশিন নিয়ে। মেকানিকাল বা যান্ত্রিক কম্পিউটারের এপিসোড এবার শেষ, শুরু নতুন কাহিনী। প্রথমে ইলেকট্রিকাল বা বৈদ্যুতিক, পরে ইলেকট্রনিক বা বৈদ্যুতিন মেশিনের পালা। এখানেই কিন্তু শেষ নয়। কম্পিউটার বদলাতেই থেকেছে, বদলাচ্ছে আজো, কিছু বদল বাজারের নিয়ম মেনে, প্রতিযোগী কোম্পানির মেশিনের সঙ্গে আমার মেশিনের বাজার দখলের লড়াই, সেই থেকে আসা বদল, আর কিছু বদল বিজ্ঞান তথা প্রযুক্তির আভ্যন্তরীন বদলের কারণে। এই দুরকম বদল আবার জল-অচল বর্গও নয়। তারা এ অন্যের সঙ্গে জড়িত। আবার বদল শুধু যন্ত্রাংশ মানে ভৌত উপাদানের স্তরেও নয়, কম্পিউটার বদলাচ্ছে তার ভৌত গঠন বা হার্ডওয়ার, এবং কাজের আভ্যন্তরীন চিন্তাপ্রক্রিয়ার গঠন বা সফটওয়ার — এই দুই নিরিখেই। এই বদল আজো চলছে। কখনো বদলটা স্বাভাবিক, দুলকি চালে, কখনো খুব নাটকীয়।

১৯৪৮ নাগাদ পদার্থবিদ ওয়াল্টার হাউজার ব্রাটেন, জন বারডিন এবং উইলিয়ম ব্রাডফোর্ড শকলে আসেন তাদের নতুন আবিষ্কার ট্রানজিস্টর নিয়ে। তার প্রকোপে আমূল এবং নাটকীয় বদলে যায় কম্পিউটারের ইতিহাস। ১৯৬০-এর দশকের শেষ দিকে আসে আইসি বা ইন্টিগ্রেটেড সার্কিট। ১৯৭০-এর দশকে আসে মাইক্রোপ্রসেসর। কিন্তু এর প্রত্যেকটা উপাদানই আমাদের আলোচনার যথাস্থানে আসবে, প্রজন্ম বা জেনারেশন থেকে জেনারেশনে কম্পিউটারের ইতিহাসের বদলের কাঠামোর সঙ্গে মিলিয়ে। সেটা নিয়ে আজ আমরা কথা বলব। প্রথমে, একদম গোড়ার কয়েকজন পথিকৃত এবং তাদের মেশিনের ছোট্ট করে উল্লেখ।

চল্লিশের দশকের মাঝামাঝি নাগাদ, মূলত যুদ্ধবাজির সরকারি তাগিদেই, প্রায় একই সঙ্গে এই ভাবনাচিন্তা শুরু হয় আমেরিকার হার্ভার্ড প্রিন্সটন পেনসিলভেনিয়া বিশ্ববিদ্যালয়ে এবং জার্মানিতে। হার্ভার্ড বিশ্ববিদ্যালয়ে হাওয়ার্ড

আইকেন, প্রিন্সটনে ইন্সটিটিউট ফর অ্যাডভান্সড স্টাডিজ-এ ভন নয়ম্যান, পেনসিলভ্যানিয়া বিশ্ববিদ্যালয়ে উইলিয়ম মশলে, এবং জার্মানির কনরাড জিউসে — এরা সবাইই এক একটা হিশেব-যন্ত্র খাড়া করেন। হাওয়ার্ড আইকেনের এই মেশিন তৈরি করে দিয়েছিল আইবিএম। হাঙ্গেরিজাত মার্কিন বৈজ্ঞানিক জন ভন নয়মানের মেশিনে প্রথম গোটা কাজটা, কাজের পদ্ধতি সহ, ভরে রাখা যেত মেমরিতে। কিন্তু সেই অর্থে প্রথম সফল কেজো কম্পিউটার যন্ত্র তৈরি হতে হতে লেগে গিয়েছিল ১৯৪৫-এ পদার্থবিদ জন মশলে আর ইঞ্জিনিয়ার জে প্রেস্পার একার্ট-এর ওই মেশিন অব্দি।

১.১।। আইকেন, হপার, মার্ক ওয়ান

মার্ক ওয়ান কম্পিটার আইকেনের নামের সঙ্গেই জড়িত, কিন্তু আইকেনের পাশাপাশি আর একজনের নাম একই ভাবে আসা উচিত, তিনি গ্রেস হপার (১৯০৬-১৯৯২)। কম্পিউটার জগতে 'বাগ' এই শব্দটা, মানে প্রোগ্রামের পোকা, অগোচরে রয়ে যাওয়া ছোট ছোট গোলযোগ, যারা প্রোগ্রাম চালানোর সময় গন্ডগোল পাকাতে থাকে, গ্রেস হপারের আমদানি। সত্যিই একটা পোকা, একটা বেচারা মথ, ঘেঁটে দিয়েছিল মার্ক ওয়ানের হার্ডওয়ার। তাই, আক্ষরিক ভাবেই, প্রথম কম্পিউটারকে প্রথমবার ডিবাগ করেন, পোকা ছাড়ান, হপার। এপিটি (APT — **A**utomatically-**P**rogrammed-**T**ool) বলে একটা ভাষা তৈরি করেন হপার এবং কোবল বলে পরে বহুলব্যবহৃত একটা ভাষা চালু করেন যারা তাদের একজন এই হপার। এপিটি ছিল একদম প্রথম যুগের একটা ভাষা। সংখ্যা দিয়ে নিয়ন্ত্রণ করা যায় এমন হার্ডওয়ার উপাদানগুলোকে প্রোগ্রাম করা যেত এপিটি-তে। মার্ক ওয়ানের মূল মানুষ ছিলেন অবশ্যই হাওয়ার্ড আইকেন (১৯০০-১৯৭৩)। আইকেন ছিলেন পদার্থবিদ এবং ইলেকট্রিকাল ইঞ্জিনিয়ার। হার্ভার্ডে তার ডক্টরেট শেষ করে ১৯৩৯-এ আইকেন গবেষণাতেই রয়ে যান। গবেষণার ফান্ড দিয়েছিল আইবিএম। এর দায়িত্বে যে তিনজনের ইঞ্জিনিয়ার টিম, তাতেই ছিলেন আইকেন আর গ্রেস।

মার্ক ওয়ান

আইকেনের মাথায় মার্ক ওয়ান জাতীয় বৈদ্যুতিক-যান্ত্রিক বা ইলেকট্রোমেকানিকাল মেশিনের ধারণাটার সূত্রপাত ঘটে ১৯৩৭-এ। তার সাত বছর পর, ১৯৪৪-এ, মার্ক ওয়ান নির্মাণের গোটা কাজটা শেষ হয়। ৪৭-এ আইকেন বানান মার্ক টু, এর পরের সংস্করণ। মার্ক ওয়ান ছিল একটা হলঘর ভর্তি মেশিন, ঘটাং ঘটাং করে তাতে ধাতুর যন্ত্রপাতি নড়ার আওয়াজ হত, ৫৫ ফুট লম্বা এবং ৮ ফুট উঁচু। পাঁচ টন ওজনের এই জরদগব বানাতে লেগেছিল সাত লক্ষ ষাট হাজার আলাদা আলাদা যন্ত্রাংশ। মার্কিন নেভির কামানের গোলা ছোড়ার কাজে ব্যবহার হয়েছিল মার্ক ওয়ান, পনেরো বছর ধরে, ১৯৫৯ অব্দি। কাগজের টেপ দিয়ে তথ্য স্থানান্তরের কাজ হত। পেপার টেপ থেকে তথ্য পড়ার জন্যে ছিল চারখানা পেপার টেপ রিডার। একটা রিডার প্রোগ্রাম মানে আদেশ জাতীয় তথ্য বা ইন্সট্রাকশন ডেটা পড়ার, অন্য তিনটে হল বিশুদ্ধ কাঁচা তথ্য বা র ডেটা পড়ার, ডেটা রিডার। পেপার টেপ ছিল একদম আইবিএম পাঞ্চড কার্ডের সমান চওড়া, মানে ৩.২৫ ইঞ্চি ।

মার্ক ওয়ান যোগ বিয়োগ গুণ ভাগ করতে পারত, এবং হিশেব প্রক্রিয়ায় পুরোনো ফলাফলকে হিশেবের মধ্যে নিয়ে আসতে পারত। মানে আগের হিশেবের আঙ্কিক তথ্য ধরে রেখে পরের হিশেবের মধ্যে নিয়ে আসতে পারত।

লগারিদম আর ট্রিগোনোমেট্রির ফাংশন ব্যবহারের জন্যে আলাদা আলাদা সাবরুটিন ছিল। দশমিকের পরে ২৩-টা অবস্থান অব্দি সংখ্যা চিবোতে পারত মার্ক ওয়ান। তথ্য ভরা এবং যান্ত্রিক ভাবে হিশেব করা — মানে কম্পিউটিং-এর কাজের জন্যে ব্যবহার হত তিন হাজার চাকা, চোদ্দশো চল্লিশ খানা ডায়াল-আকৃতির গোল সুইচ, এবং গোটা মেশিনে ব্যবহৃত মোট তারের পরিমাণ ছিল পাঁচশো মাইল। ইলেকট্রিক রিলে ব্যবহৃত হয়েছিল। সমস্ত আউটপুট ফুটিয়ে তুলত একটা ইলেকট্রিক টাইপরাইটারে। একটা গুণ করতে লাগত তিন থেকে ছয় সেকেন্ড। লগ বা ট্রিগোনোমেট্রির হিশেব করতে লাগত মোটামুটি এক মিনিট বা তার বেশি। গোটা যন্ত্রের যান্ত্রিক কাজটা করত মেশিন বরাবর লম্বায় ডান্ডায় লাগানো একটা চার অশ্বশক্তির ইঞ্জিন। আজকের আপনাকে ধরুন তুলে নিয়ে ওই হলঘরে ঢুকিয়ে দেওয়া হল, এবং জিগেশ করা হল, যন্ত্রটা কী বলোতো?

১.২।। জন ভন নয়মান এবং ফলিত গণিত

জন ভন নয়মানের জন্ম ২৮ ডিসেম্বর, ১৯০৩-এ, হাঙ্গেরির বুদাপেস্তে, মৃত্যু ৮ ফেব্রুয়ারি, ১৯৫৭, ওয়াশিংটন ডিসিতে। খোদার কুদরতে নয়মানের প্রতিভার চিহ্ন ফুটে উঠতে শুরু করেছিল তার ছ-বছর বয়েসেরও আগে থেকে, যখন ধ্রুপদী গ্রীক ভাষায় বাবার সঙ্গে জোকস বিনিময় শুরু করে ইয়ানুস, মার্কিন ভাষায় জনি। এবং বাড়ির লোক স্তম্ভিত হয়ে যেত বাচ্চা জনির আস্ত আস্ত টেলিফোন ডিরেক্টরি মুখস্থ রাখার ক্ষমতায়। ১৯২৩ অব্দি বার্লিন বিশ্ববিদ্যালয়ে রসায়ন পড়ার পর নয়মান যান জুরিখ, সেখানে কেমিকাল ইঞ্জিনিয়ারের ডিপ্লোমা পান। জুরিখ থেকেই শুরু নয়মানের গণিতপ্রেমের। বুদাপেস্ত বিশ্ববিদ্যালয় থেকে গণিতে ডক্টরেট করেন ১৯২৬-এ, সেট থিয়োরির উপর তার তত্ত্বে। পোস্ট-ডক্টরাল করতে যান সেই রূপকথা হয়ে যাওয়া গোটিংগেন বিশ্ববিদ্যালয়ে, জার্মানির গোটিংগেনকে রূপকথা করে তুলেছিলেন যারা তাদের প্রমুখ ডেভিড হিলবার্টের ছাত্র হয়ে। ডেভিড হিলবার্ট (১৮৬২-১৯৪৩) সেই গণিতজ্ঞ-দার্শনিক, ১৮৯৯-এ বেরোনো যার 'দি ফাউন্ডেশনস অফ জিয়োমেট্রি' ইউক্লিডের জ্যামিতির জায়গায় একটা নতুন জ্যামিতিকে নিয়ে আসার কাজটাকে সম্পূর্ণ করেছিল, ইউক্লিডের থেকে সম্পূর্ণ স্বতন্ত্র একটা স্বতঃসিদ্ধমালায় নির্মিত একটা নতুন জ্যামিতি — যার উল্লেখ করেছিলাম তিন নম্বর দিনে।

তদ্দিনে নয়মান স্টার থেকে সুপারস্টার হওয়ার পথে। জ্যামিতি এবং টোপোলজিতে তুমুল কাজ করা অধ্যাপক অসওয়াল্ড ওয়েবলেন তাকে ডেকে নেন কোয়ান্টাম থিওরির কাজে। ১৯৩১-এ নয়মান প্রিন্সটন বিশ্ববিদ্যালয়ে অধ্যাপক হিশেবে যোগ দেন। বেধড়ক উজ্জ্বল ছাত্র না-হলে নয়মানের কাছে পড়া প্রায় অসম্ভব ছিল। নয়মানের একটা প্রিয় খেলা ছিল তার ছাত্রদের কাছে সন্ত্রাস — একটা বিতিকিচ্ছিরি রকমের জটিল সমীকরণ বোর্ডের এককোণে বিদ্যুৎবেগে সমাধান করেই, ছেলেমেয়েরা লেখার আগে সেটা মুছে ফেলা, তারপর সেই সমাধানের স্টেপগুলো নিয়ে আলোচনা। নাইটক্লাবের ক্যাবারে পার্টির সার্কিটে এমন ছেদহীন উপস্থিতি ছিল তার যে তাকে ক্যাবারে-দুনিয়ারই নাগরিক বলে উল্লেখ করা আছে নয়মান বিষয়ে এক স্মৃতিচারণে। আর আদিরসায়নে নয়মানের বিশেষজ্ঞতার কথা তো আগেই বলেছি। নয়মানের বিরাট বিরাট কাজ আছে মেজার থিওরি, স্ট্যাটিস্টিকাল মেকানিক্স, কোয়ান্টাম মেকানিক্স-এ, টোপোলজিতে, গেম থিয়োরিতে, নন-লিনিয়ার পার্শিয়াল ডিফারেনশিয়াল ইকুয়েশনে। জীবনের শেষ বছরগুলোর একটা বড় সময় কাটিয়েছিলেন সাইবারনেটিক্সে, অটোমাটার কাজে। অটোমাটা হল অটোমেশনের

বহুবচন, মানে রোবট জাতীয় যে কোনো স্বয়ংক্রিয় যন্ত্র। এদের আভ্যন্তরীন গঠনের গণিতটাই হল অটোমাটা থিওরির ভিত্তি। এখানেই কাজ করেছেন ভন নয়মান।

১৯৩০-এর দশকে হাইড্রোডিনামিক্স বা তরলগতিবিদ্যা নিয়ে কাজ শুরু করেন নয়মান। অভ্যস্ত গাণিতিক কায়দাকানুন দিয়ে এর সমীকরণগুলোকে কব্জা করা সম্ভব ছিলনা। নয়মান তখন এদের নিউমেরিকাল অ্যানালিসিস দিয়ে সাইজ করার ধরার কথা ভাবেন। নিউমেরিকাল অ্যানালিসিস বলতে, এক কথায়, অতিসরলীকৃত রকমে — গণিত দিয়ে একটা সমীকরণকে সরাসরি সমাধান করার চেষ্টা না-করে, পরপর অনেকগুলো ধাপে একটা সমীকরণের একটা অ্যাপ্রক্সিমেট বা মোটামুটি সমাধানের আঙ্কিক মানে পৌঁছনোর চেষ্টা করা। এর একটা কারণ, বহু সমীকরণ গাণিতিক রকমে সমাধান করা যায়না, বা বহুসময় অকল্পনীয় জটিলতা জড়িয়ে থাকে তার সঙ্গে। কাজের প্রয়োজনটা মিটে যাওয়াই এখানে বড় কথা, দশমিকের পরে কয়েক ঘর অব্দি সমাধানটার মান পেলেই যদি চলে যায়, তাহলেই হল। তাত্ত্বিক ভিত্তিটা এখানে এই যে ইনফিনিট বা অনন্ত-সংখ্যক ধাপের পর এই পাটীগাণিতিক মানের অ্যাপ্রক্সিমেট বা মোটামুটি সমাধানটা মূল সমাধানের সঙ্গে বেমালুম মিলে যাবে। কিন্তু সমীকরণটাকে সরাসরি সমাধান না করে এই অ্যাপ্রক্সিমেট আঙ্কিক মানের ভিত্তিতে তাকে ইন্টারপ্রিট বা ব্যাখ্যা করার এই বহুল ব্যবহৃত পদ্ধতিটার ঝঞ্ঝাট এটাই যে একটা বিপুল পরিমাণ হিশেব কষার, কষে যাওয়ার দরকার পড়ে, প্রতিটি ধাপে। নয়মানের তাই দরকার ছিল একটা কম্পিউটারের। নয়মান, স্বাভাবিকভাবেই, একটা কম্পিউটারের গঠন কী হতে পারে তাই নিয়ে ভাবনাচিন্তা শুরু করলেন। তার ফল হল যে ছকটা সেটা ভন নয়মান আর্কিটেকচার বলেই পরিচিত। আমরা সেই শূন্য থেকে যে কাজের ভিত্তিতে কম্পিউটারের ছকটা দিয়েছিলাম — সেই বিভাগটা নয়মানই করেছিলেন প্রথম — অ্যারিথমেটিক অ্যান্ড লজিকাল ইউনিট, কন্ট্রোল ইউনিট, বিরাট আকারের একটা কেন্দ্রীয় মেমরি এবং ইনপুট/আউটপুট ডিভাইস।

এবং এর সঙ্গে সবচেয়ে গুরুত্বপূর্ণ জায়গাটা দিলেন — তুলে রাখা বা স্টোরড প্রোগ্রামের ধারণা। মানে, যে আদেশগুলো কম্পিউটারে ভরে দিতে হবে, তার তথ্যের পাশাপাশি, এটা অনুযায়ীই চলবে বা কাজ করবে মেশিন। মানে যে নিয়ন্ত্রণটা এর আগে অব্দি মানবসভ্যতার ইতিহাসে কোনো যন্ত্রকে মানুষ কোনোদিন বকলমা করতে পারেনি, তার সুযোগটা এল এইবার। তিন নম্বর দিনে আমরা বলেছিলাম হিশেব কষার শারীরিক ভার ভাগ করার জায়গা থেকে এসেছিল অ্যাবাকাস। এবার, হিশেব কষার গাণিতিক কাজের নিয়ন্ত্রণের মানসিক ভারটা তুলে দেওয়ার সুযোগ এল — কম্পিউটার নামক যন্ত্রের হাতে। যে নিয়ন্ত্রণটা নিজেদের মাথায় রাখতে হত, এখন থেকে সেটা তোলা থাকবে মেশিনের মেমরিতে — এবং এটার নামই প্রোগ্রাম। প্রয়োজনীয় আদেশগুলোকে একসঙ্গে এইভাবে তুলে রাখা বা স্টোর করে রাখতে পারা মানে প্রত্যেকবার মেশিন চালানোর আগে নতুন করে মেশিনকে আদেশ ঢোকানোর প্রয়োজন হবেনা। আর তখনকার দিনে নতুন আদেশমালা মানে কী ভাবুন — বড় বড় নতুন নতুন কানেকশন তৈরি করা নতুন নতুন তার জুড়ে জুড়ে, পরে আমরা দেখব, প্লাগবোর্ড দিয়ে। আমাদের কাছে আজকে গোটা ব্যাপারটা খুবই সাদামাঠা লাগছে। তখনকার কম্পিউটারের দুনিয়ায় এটা একটা বিরাট পদক্ষেপ। দার্শনিক ভাবেও, প্রযুক্তিগত রকমেও।

তার সময়কার অন্য অনেকের থেকে আরো অনেক বেশি করে কম্পিউটারকে একটা গাণিতিক অস্ত্র হিশেবে দেখেছিলেন ভন নয়মান। তার চিন্তনগত তাৎপর্যটা ধরতে পেরেছিলেন। বিদ্যাচর্চা তথা সভ্যতার আগামী ইতিহাস জুড়ে ফলিত গণিত বা অ্যাপ্লায়েড ম্যাথামেটিক্স-এর যাবতীয় সমস্যা সমাধানের একটা প্রবল উপায় উপায় হিশেবে কম্পিউটারের সম্ভাবনাটা আগে থেকেই ভেবে উঠতে পেরেছিলেন ভন নয়মান। হাইড্রোডিনামিক্স, ব্যালিস্টিক্স বা গোলাগুলি-মিসাইল ইত্যাদি ছোঁড়ার বিষয়ে বিজ্ঞান, মেটেরিয়োলজি বা আবহাওয়া-বিজ্ঞান, গেম থিওরি এবং রাশিশাস্ত্র বা স্ট্যাটিস্টিক্স — এই সবগুলোর উপরেই নয়মানের অগাধ এবং তুলনাহীন পাণ্ডিত্য প্রবলভাবে ব্যবহৃত হয়েছিল দ্বিতীয় বিশ্বযুদ্ধের সময়, যুদ্ধবিজ্ঞানের প্রায় প্রতিটি স্তরে। হয়তো কোনো মানে নেই, তবু এই শেষ বাক্যটা লিখতে আমার বেশ খারাপ লাগল।

নয়মান প্রথম যে কম্পিউটারের সঙ্গে সম্পৃক্ত হন সেগুলো হল মুর স্কুলের এডভ্যাক তথা এনিয়াক, যার কথায় আমরা এখুনি আসছি এবং আইকেনের ওই ম্যাক ওয়ান। ওই সময়ই নয়মান আগ্রহী হয়ে ওঠেন কম্পিউটারের রিলে সিস্টেম নিয়ে। যুদ্ধের শেষ দিক থেকে শুরু করে নয়মান বানান তার আইএএস বা ইনস্টিটিউট অফ অ্যাডভান্সড স্টাডিজ-এ তার নিজের কম্পিউটার, পরে যার মডেলে তৈরি মেশিনগুলো পরিচিত হয়েছিল 'সুপারকম্পিউটার'

বলে। পরবর্তীযুগের কম্পিউটার বিবর্তনের উপর যার বিরাট প্রভাব। প্রতিভা এবং অবদানের নিরিখে কম্পিউটার বিজ্ঞানের এই যুগটায় নয়মানের তুলনা বোধহয় শ্যানন আর ট্যুরিং ছাড়া আর কারো সঙ্গেই করা যায়না।

১.৩।। মশলে, একার্ট, এনিয়াক

এনিয়াক (ENIAC — **E**lectronic-**N**umerical-**I**ntegrator-**A**nd-**C**omputer) ছিল মার্ক ওয়ানেরও বাড়া, ওজনে তিরিশ টন। এতে ছিল ১৮০০০ ভালভ বা ভ্যাকুয়াম টিউব। চালানোর সময় বিদ্যুৎ লাগত ১৭৪ কিলোওয়াট এবং জায়গা লাগত পঞ্চাশ ফুট বাই তিরিশ ফুট। শ্যানন কাজ করেছিলেন যে ডিফারেনশিয়াল অ্যানালাইজারে, তিন নম্বর দিনে, লজিক সার্কিট উদ্ভাবনের সূত্রে আমরা যে মেশিনের কথা বলেছিলাম, তার থেকেই, তার উত্তরাধিকারী ছিল এই এনিয়াক। যুদ্ধের আগে, মূলত দূরপাল্লার গোলন্দাজির কাজে সহায়তার কারণে এনিয়াক বানানো হয়েছিল। এই সময়ের এই কম্পিউটার প্রকৌশলের এবং গোলন্দাজি বিজ্ঞানের গোড়ার দিককার সময় নিয়ে কিছু মজার গল্প আছে রিচার্ড ফেইনম্যানের আত্মজৈবনিক লেখা 'শিওরলি ইউ আর জোকিং মিস্টার ফেইনম্যান' বইটাতে, দু-একটা দেওয়াও যেত, কিন্তু জিএলটির কেয়া বইটা নিয়ে গেছে, কাছে নেই। ফেইনম্যান এই সময় কিছুদিন অর্ডন্যান্সে চাকরি করেন, সেই সময়টার নানা মজার অভিজ্ঞতা।

কামান থেকে গোলা ছোঁড়ার সময়, শেষ অব্দি কতটা দূর যাবে একটা গোলা, সেটা নির্ভর করে কতটা জোরে ছোঁড়া হচ্ছে, কত ডিগ্রিতে বেঁকানো আছে নলটা, গোলাটার আকার আকৃতি এগুলো ছাড়াও আবহাওয়ার অবস্থা তখন কী, বাতাসের গতি, বাতাসে আর্দ্রতা কতটা, সমুদ্রপৃষ্ঠ থেকে কতটা উঁচুতে আছে কামানটা, বায়ুমণ্ডলের তাপমাত্রা কী, এবং ভূমির উঁচুনিচু ইত্যাদি নানা কারণে সেখানে অভিকর্ষের অবস্থা — এইরকম অনেককিছুর উপর। গণিতের ভাষায় একে বলে প্যারামিটার বা নিয়ন্তা। এবার ভাবুন, একটা কামান মানেই কামানের সঙ্গে বিভিন্ন ধরনের মানের তালিকায় ভর্তি একটা বই। গোলা ছোঁড়ার সময় এই সমস্ত প্যারামিটার বা নিয়ন্তাগুলোর মানের তালিকা, তার সঙ্গে মিলিয়ে গোলা পৌঁছনোর সম্ভাব্য দূরত্ব। সভ্যতার হদ্দমুদ্দ একেবারে, সাক্ষর গাণিতিক সুচারু খুনোখুনি। বই পড়ো, গুলি মারো। কামান বদলানো গোলা বদলানো মানেই বই বদলানো।

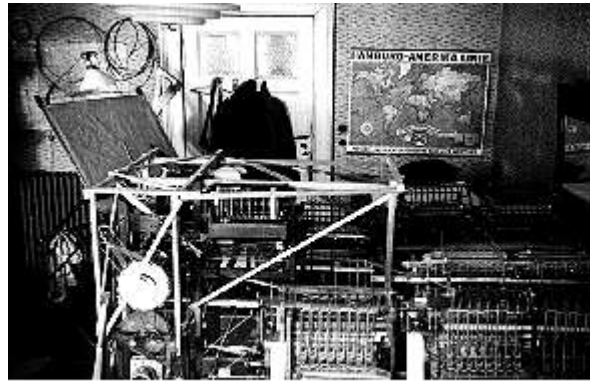
আমেরিকায় এই গোলাবিজ্ঞানের কাজটা, প্যারামিটার সহ সম্ভাব্য দূরত্বের তালিকার বই লেখা যার একটা অংশ, প্যারামিটারের মানগুলো খুঁজে পাওয়া যার মূল গবেষণা — এর প্রায় গোটাটাই হত মেরিল্যান্ডের আবেরডিনে আর্মি অর্ডন্যান্স ব্যালিস্টিক রিসার্চ ল্যাবরেটরিতে। নতুন যুদ্ধের প্রস্তুতি মানেই রাশি রাশি নতুন হিশেব এবং নতুন বই। হিশেবের কাজ করার জন্যে ছিল ডিফারেনশিয়াল অ্যানালাইজার বা হাতে ডান্ডা ঘোরানো বিভিন্ন যান্ত্রিক ক্যালকুলেটর, ফ্যাসিট ইত্যাদি, যেগুলোর কথা আগেই বলেছি, বা একদম বাংলা রকমে হাতে করে। কিন্তু এইভাবে এসব করা মানেই খুব কম হলেও, একটা নতুন সেটের জন্যে বেশ কয়েক মাস লেগে যাওয়া। এই কাজের মূল দায়িত্বটা তাই দেওয়া হল পেনসিলভানিয়া বিশ্ববিদ্যালয়ে মুর স্কুল অফ ইলেকট্রিকাল ইঞ্জিনিয়ারিং-এর উপর। মুর স্কুলে এই সময় ছিলেন জন উইলিয়ম মশলে (১৯১৯-১৯৮০) এবং জন প্রস্পার একার্ট (১৯১৯-১৯৯৫)।

আবহাওয়া বিদ্যা এবং আবহাওয়ার পূর্বাভাষের ব্যাপারে এর আগে থেকেই আগ্রহী ছিলেন মশলে। সেই কাজে মূল প্রতিবন্ধকই ছিল বড় বড় একঘেয়ে শেষহীন জাম্বো হিশেব। তাদের সহজে চটকানোর স্বার্থে কম্পিউটারের বিষয়ে আগে থেকেই আগ্রহী ছিলেন মশলে। এই রকম সময়ে মশলের কানে আসে আইওয়া স্টেট কলেজে এবিসি বা

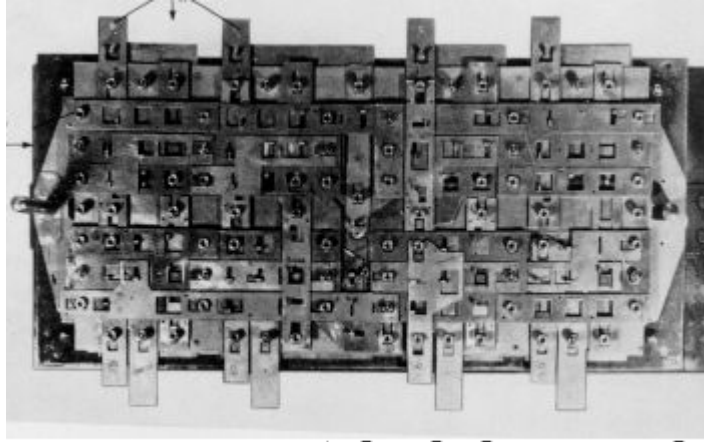
আটানাসফক্ট বেরি কম্পিউটারের কথা। '৪১-এ গিয়ে এই এবিসি দেখে এবং শিখে আসেন মশলে। এ বি সি শেখার পর এবার আর বাক্য লিখতে কতক্ষণ, বিয়াল্লিশের বসন্তে মশলে প্রস্তাব করলেন একটা বড় মাপের ইলেকট্রনিক ডিজিটাল কম্পিউটারের। কিন্তু বিশ্ববিদ্যালয়ের দাদারা, বিগ বসরা, এতে মুগ্ধ হলেন না, এবং ধামাচাপা পড়ে গেল এই প্রোজেক্ট। শেষ অব্দি, ফের প্রস্তাবের পর, '৪৩-এ এসে কাজ শুরু হল এনিয়াকের। মূল ইঞ্জিনিয়ার একার্ট এবং বিশেষজ্ঞ মশলে। পুরোদস্তুর কাজের মত মেশিন হয়ে এনিয়াক দাঁড়িয়ে উঠতে লেগে গেছিল '৪৬ অব্দি। মোগাম্বো খুশ হয়েছিলেন।

তার সময়ের অন্য মেশিনের তুলনায় এনিয়াকের গতি ছিল অবিশ্বাস্য রকমের বেশি। যোগ করতে লাগত ২০০ মাইক্রোসেকেন্ড। এক সেকেন্ডের দশলক্ষ ভাগের এক ভাগে হয় এক মাইক্রোসেকেন্ড। গুণ করতে লাগত ২.৬ মিলিসেকেন্ড, মিলিসেকেন্ড মানে এক সেকেন্ডের এক হাজার ভাগের এক ভাগ। বর্গমূল বার করতে লাগত ২৫ মিলিসেকেন্ড। বর্গমূল বার করায় এই বাড়তি গড়িমসির কারণ কোনো সংখ্যার বর্গ বার করার যে বিশেষ অ্যালগরিদমটা এনিয়াক ব্যবহার করত সেটার বিদঘুটেপনা। তাতে শুধু যোগ আর বিয়োগ করেই কাজ সারা যেত বটে, কিন্তু বড় বড় সংখ্যার বেলায় যে পরিমাণ যোগ আর বিয়োগ করতে হত সেটা রীতিমত অশ্লীল। অ্যালগরিদম বা ক্রিয়াপদ্ধতিটাও খুব মজার। কিন্তু নাম্বার সিস্টেমটা আগে একটু বলে না-নেওয়ায় সেটায় যেতে পারলাম না।

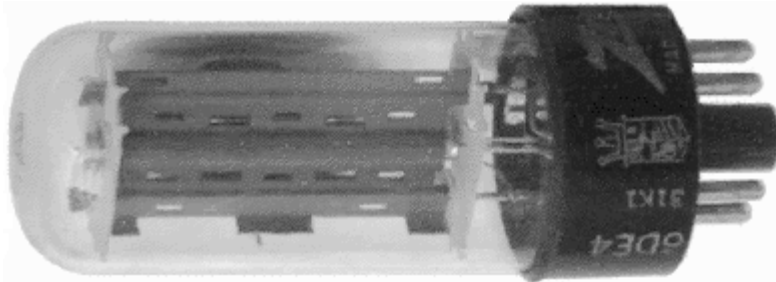
আর মূল যে কাজের এনি আঁক কষে দেওয়ার জন্যে এনিয়াককে এনে ফেলা হয়েছিল এই ধরাধামে, সেই যুদ্ধু ডিপার্টমেন্টের মহাপ্রভুরাও খুশি হয়েছিলেন তাদের এই ওজনদার দাসের কাজকর্মে। সবকটা নিয়ন্তা বা প্যারামিটার, ওই যে প্রভাবক বিষয়গুলো আমরা একটু আগে বললাম, তাদের মানগুলো দিয়ে দেওয়ার পর একটা গোলার গতিপথ বা ট্রাজেক্টরি কষতে এনিয়াকের লাগত তিরিশ সেকেন্ড, যে কাজে শ্যাননের ডিফারেনশিয়াল অ্যানালাইজারের লাগত পনেরো মিনিট, ক্যালকুলেটর হাতে একটা মানবকের যাতে লাগবে কমসেকম বিশ ঘন্টা। এদিকে গোলাটার গড়ে এক মিনিট লাগত ওই দূরত্বটা পেরিয়ে তার চাঁদমারিতে পৌঁছতে, তার মানে গোলার গতির চেয়ে গোলার গতির হিশেবের গতি ছিল এনিয়াকের। তাই এনিয়াককে ভালোবেসে বলাই যায়, গতিতে তুমি বজ্রতর, হে এনিয়াক। এনিয়াক কাজ করেছিল ১৯৫৫ অব্দি। আশি হাজার দুশো তেইশ ঘন্টা কাজ করার পর, হয়তো গান্ধীর অহিংসায় দীক্ষিত হওয়ার আশাতেই, ওয়ার ডিপার্টমেন্টের এই মেশিন শেষ বারের মত অফ করে দেওয়া হয়, ১৯৫৫-র ২ অক্টোবর, গান্ধীজয়ন্তীর দিন।

১.৪।। কনরাড জুসে, জেড সিরিজ, প্লানকালকুল

নিজের ঘরে বানিয়ে তোলা জেড ওয়ান

এর আগে যে তিনটে উদাহরণ দিলাম, হার্ভার্ডে আইকেন, প্রিন্সটনে নয়মান, পেনসিলভানিয়ায় মশলে — এই তিনটেই আমেরিকায়। এই প্রথম যুগের কম্পিউটারগুলোয় আমেরিকার বাইরে একটাই উদাহরণ, জার্মানিতে কনরাড জুসে (জন্ম বাইশে জুন, ১৯১০)। তাই, তুলনামূলক ভাবে অনেক কম উল্লেখ বা প্রচার হয় জেড-সিরিজের, পৃথিবীর মুখের ভাষা আর বুকের চিন্তাকে নির্ণয় করে দেয় আমেরিকা। অথচ, সেদিক দিয়ে দেখলে প্রথম কম্পিউটারের সম্মান পাওয়া উচিত জুসের, কারণ জুসের জেড সিরিজের প্রথম মেশিন জেড ওয়ান তৈরি হয়েছিল ১৯৩৬ থেকে ১৯৩৮-এর মধ্যে। আর আইকেনের মার্ক ওয়ান তৈরি হতে লেগেছিল ১৯৩৯ থেকে ৪৪।

জুসে জেড ওয়ান বানান নিজেদের বসার ঘরে, প্রতিটি শব্দ ২২ বিটের এমন ৬৪-টি শব্দ নিজের স্মৃতিতে রাখতে পারত জেড ওয়ান। একে প্রোগ্রাম করা যেত এবং কাজ করত বাইনারিতে। জেড ওয়ান মেশিনে কোনো বৈদ্যুতিক রিলে ব্যবহৃত হয়নি, সেই কাজগুলো গোটাটাই করা হয়েছিল ধাতুর পাত জুড়ে জুড়ে, ছবিতে দেখুন। জিগ-স করাত দিয়ে কেটে কেটে জুসে আর তার বন্ধুরা নিজেদের হাতে বানিয়েছিল এই পাতগুলো। একমাত্র বৈদ্যুতিক যন্ত্রাংশ ছিল একটা মোটর যার দৌলতে একটা এক হার্জের ক্লক-ফ্রিকোয়েন্সি তৈরি হত মেশিনে। প্রোগ্রাম ঢোকানো হত একটা পাঞ্চ টেপ আর পাঞ্চ টেপ রিডার দিয়ে। শূন্য নম্বর দিনে আমাদের আনা সেই সরলতম আর্কিটেকচারের নিরিখে ভাবতে গেলে, পরিস্কার আলাদা করা যেত নিয়ন্ত্রণ উপাদান বা কন্ট্রোল ইউনিটকে এই পাঞ্চ টেপ রিডার থেকে। আলাদা করা যেত এর অ্যারিথমেটিক ইউনিট বা প্রসেসিং অংশ এবং স্মৃতি বা মেমরিকেও এবং ইনপুট-আউটপুট ডিভাইসকেও। চমৎকার করে এই গোটাটা বোঝানো আছে কনরাড জুসের ছেলে হোর্স্ট জুসের লেখা বই ‘দি লাইফ অ্যান্ড ওয়ার্ক অফ কনরাড জুসে’ নামে একটা বইয়ে, গোটা বইটাই নেটে পাওয়া যায়। www.epemag.com সাইট থেকে। কনরাড জুসের এই জেড সিরিজে জেড টু জেড থ্রি ইত্যাদি আরো সংযোজন এসেছিল। গোটা গল্পটা বেশ ইন্টারেস্টিং।

পাতলা ধাতুর পাত জুড়ে জুড়ে তৈরি হয়েছিল হিশেব করার যন্ত্রপাতি

যুদ্ধের সময় বোম পড়ে সম্পূর্ণ নিশ্চিহ্ন হয়ে যায় জেড ওয়ান। কিন্তু নাজি সরকার তার কাজের কোনো পৃষ্ঠপোষকতা করেনি, জুসে পালিয়ে আসেন জুরিখে, জেড-থ্রির কিছু খণ্ডাংশ নিয়ে। শুধু কম্পিউটারই নয়, একটা ভীষণ প্রাথমিক স্তরের অপারেটিং সিস্টেম কাম হাইলেভেল ল্যাংগুয়েজও বানান জুসে, তার নাম ‘প্লানকালকুল’। ডেসিমাল বা দশমিক পদ্ধতিতে সাতাশটা অব্দি সংখ্যা নিয়ে কাজ করতে পারা আডা-ব্যাবেজের অসমাপ্ত অ্যানালিটিকাল ইঞ্জিনের কথা জুসে জানতেন না। স্বতন্ত্র ভাবে জুসে নিজেই একটা প্রোগ্রামিং কাঠামো ভেবে বার করেন, যার ফলাফল প্লানকালকুল।

ভ্যাকুয়াম টিউব বা ভালভের ভিতর থাকে এক সেট ধাতব ইলেকট্রোড বা তড়িৎদ্বার, আর এদের সবাইকে ধরে রাখার আর বিদ্যুৎ যোগাযোগের জন্যে একটা তারের কাঠামো বা গ্রিড। এই গোটাটা ভরা থাকে একটা বায়ুশূন্য কাঁচের বা ধাতুর টিউবে।

ভ্যাকুয়াম টিউব বা ভালভ

গ্রিডের শরীরে ভোল্টেজ বা তড়িৎ-বিভবের মানই ইলেকট্রোডগুলোর ভিতর তড়িৎপ্রবাহকে নিয়ন্ত্রণ করে। ভালভ ব্যবহার করা হত অ্যাম্পলিফিকেশন বা পরিবর্ধনের কাজে, বা সার্কিটে তড়িৎপ্রবাহ নিয়ন্ত্রণের কাজে, এখনও ব্যবহার হয়, ক্যাথোড রে

টিউবে, বা ভয়ানক উঁচু মাত্রার বিদ্যুৎ যেখানে ব্যবহার হয়। এর মার্কিন নাম ভ্যাকুয়াম টিউব, ব্রিটিশ নাম ভালভ। আমাদের, মেকলের অবৈধ বাচ্চাদের, দুটোই জানার কথা।

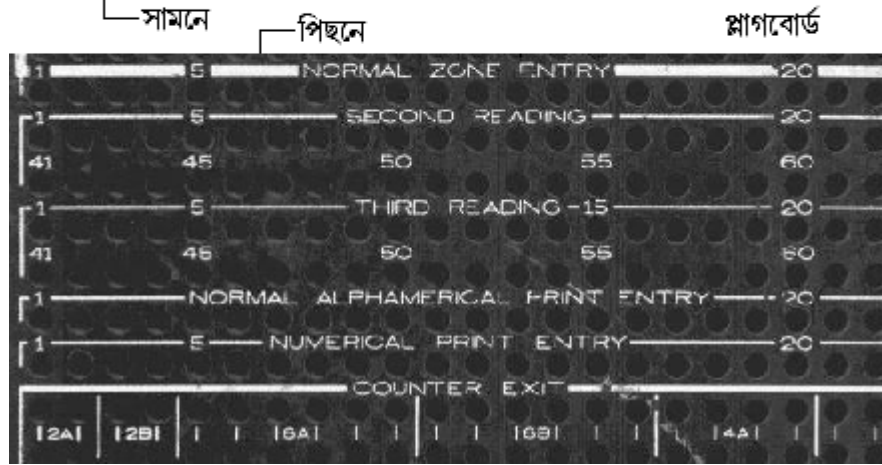
২।। প্রথম জেনারেশন কম্পিউটার — ১৯৪৫-৫৫

এই আইকেন জুসে নয়মান মশলেদের কাঁধে ভর করে পৃথিবীতে দেখা দিলেন আধুনিক ডিজিটাল কম্পিউটারের প্রথম প্রজন্ম। বা ফার্স্ট জেনারেশন। এই স্তরের কম্পিউটারগুলোকে এক কথায় ডাকা যায় ভ্যাকুয়াম টিউব আর প্লাগবোর্ডের মেশিন বলে। তখনো ইলেকট্রনিক রিলে ব্যবস্থা আসেনি — শ্লথগতি যান্ত্রিক রিলেই ব্যবহৃত হয়েছিল। পরে এই রিলেগুলোর জায়গায় আনা হয় ভ্যাকুয়াম টিউব। এই যুগের কম্পিউটারগুলোর আকার ছিল বিপুল। মার্ক ওয়ান বা এনিয়াক গোছের এদের পূর্বপুরুষদের সাইজ তো এইমাত্রই দেখলেন। আস্ত আস্ত ঘর হলঘর ভরে যেত এক একটা কম্পিউটারের ভিতরের হাজার হাজার ভ্যাকুয়াম টিউবে। ভ্যাকুয়াম টিউব থেকে পরবর্তী যুগের ট্রানজিস্টরে রূপান্তর যেমন একটা নাটকীয় বদল, এর আগের অবস্থান থেকে ভ্যাকুয়াম টিউবও একটা নাটকীয় বদল ছিল।

অথচ এই সময়কার মেশিনগুলোর এই জগদ্দল আয়তনের পরেও, আজকের পৃথিবীর সবচেয়ে সস্তা কম্পিউটারও এদের চেয়ে অযুত গুণ দ্রুত চলে। এই সময়ের কম্পিউটারগুলো চলত আজকের আমাদের কাছে সম্পূর্ণ অপরিচিত এক কায়দায়। এক একটা কম্পিউটার মানেই তখন এক এক সেট লোক — যারা মেশিনটা বানিয়েছে, যারা মেশিনটা চালায়, যারা প্রোগ্রাম করে এবং যারা গোটা ব্যাপারটার রক্ষণাবেক্ষণ করে। এদের বাইরে অন্য সবার কাছে কম্পিউটার তখন একটা অবোধ্য জগত। এই মেশিনের প্রোগ্রামগুলোও লেখা হত মেশিন ল্যাংগুয়েজে। এমনকি অ্যাসেম্বলি ল্যাংগুয়েজও তখন আসেনি।

অ্যাসেম্বলি ভাষা মানে একদম নিম্নতম স্তরে কম্পিউটার আর্কিটেকচারকে নির্দেশ দেওয়ার জন্যে তৈরি এক ধরনের একটা ভাষা, যেমন অনেকটা উপরের স্তরের একটা ভাষা হল সি। একটা ভাষা মানে কিছু বিশেষ অর্থবাচক শব্দ এবং কিছু নিয়মের সমষ্টি। বাংলা ইংরিজি যে কোনো ভাষা মানেই তাই। এই বললে আমি এই বুঝব, ওই বললে ওই, এবং এই এই জিনিষকে এই এই ভাবে বলা যাবে। কিন্তু বাস্তব জীবনের যে কোনো ভাষার একটা ইতিহাস থাকে, কী ভাবে এই বিশেষ অর্থবাচক শব্দ আর বিশেষ নিয়মগুলো গজিয়ে উঠল। ব্যক্তি মানুষের কোনো একক বা সমষ্টি সেখানে সিদ্ধান্ত নেয়না, সিদ্ধান্ত নেয় জ্যান্ত ইতিহাস। কিন্তু কম্পিউটার ভাষারা সেই অর্থে নির্মিত। বিশেষ কিছু কাজের উদ্দেশ্যে বানিয়ে তোলা হয়। সেখানে উদ্দেশ্যটা থাকে এমন ভাবে অর্থপূর্ণ শব্দের তালিকা আর নিয়মগুলো বানিয়ে তোলা, যাতে কাজগুলো করা সহজ হয়। যাতে, এই নিয়মগুলোকে ব্যবহার করে ক্রমে জটিল জটিলতর নির্দেশও সহজে দেওয়া যায়। আমাদের রোজকার ব্যবহারের ভাষার সঙ্গেও এই অর্থে একটা কাঠামোগত মিল আছে এর। কম্পিউটার ভাষার এক একটা কি-ওয়ার্ড হল আমাদের ভাষার এক একটা জার্গনের মত। ধরুন, এই যে আমরা এখন মুর্হুমুহূ 'কিবোর্ড' শব্দটা ব্যবহার করতে পারছি কারণ ওই শব্দের পিছনে গোটা অর্থ বা ধারণা বা ছবিটা আমাদের কাছে প্রদত্ত। এবার এদের ব্যবহার করে জটিলতর ধারণাগুলোকেও আমরা পৌঁছে দিতে পারছি। এর আবার কিছু নিয়ম আছে, যেমন, আমরা কিবোর্ড সরিয়ে রাখি বা চাবি-টিপি কিন্তু কাপে ঢেলে রাখিনা বা উড়িয়ে দিই-না। কিন্তু খেয়াল করুন, এখানে পারা বা না-পারাটা বাস্তবতা থেকে এসেছে, আমাদের কিছু করার নেই, একটা চৌকো জড় পদার্থকে ঢালা বা ওড়ানো যায়না, কিন্তু কম্পিউটার ভাষার বেলায় এটাও বানিয়ে তোলা, ভাষা উদ্ভাবক বা প্রোগ্রামচিরা মিলে ঠিক করে নেওয়া। ঠিক একই ভাবে, সি ভাষার কিছু কি-ওয়ার্ড আছে, যেমন, শূন্য নম্বর দিনে আমাদের সেই গাধা-বাচক প্রোগ্রামে 'int', 'return' ইত্যাদি। কিছু নির্দিষ্ট নিয়ম আছে, যা মেনে নির্দেশগুলোকে লিখতে হয়। নিয়মও আছে, ওই কোডটার একটা ';' তুলে নিলেও কোডটা আর ব্যবহার করা যেতনা।

একটা কম্পিউটারে একটা ভাষা ব্যবহার করা যায়, মানে, ওই ভাষার কি-ওয়ার্ডগুলো, নিয়ম-কানুন, ব্যাকরণ মেশিনটার মধ্যে ভরা আছে। সেই নিয়মকানুন মেনে কথা বললে, কী বললে কী করতে হবে এই সরলতর নির্দেশগুলো তার মধ্যে আগে থেকেই ভরা আছে। এই অ্যাসেম্বলি ভাষা আবার এতটাই সরল সাদামাঠা এবং আপাতদৃষ্টিতে অঙ্কের লাইনের মত দেখতে যে তাকে ভাষার চেয়ে বাচন বলাই শ্রেয়। সেই অ্যাসেম্বলি ভাষাও ভরা

ছিলনা এই যুগের এই মেশিনগুলোয়। প্রোগ্রামিং ল্যাংগুয়েজ বা অপারেটিং সিস্টেম — ওসবের কেউ নামই শোনেনি। কম্পিউটারকে ন্যূনতম আদেশটাও দিতে হত একদম সরাসরি নিম্নতম আর্কিটেকচার যে আদেশ বুঝতে এবং পালন করতে পারে সেই আদেশের এককে।

একটা প্রোগ্রাম চালানো মানে প্রোগ্রামচি এসে একটা নির্দিষ্ট সময়ের জন্যে বুক করে নিত মেশিনটাকে, তারপর চলে যেত সেই রুদ্ধ কক্ষে — বিনা কারণে, বিনা অনুমতিতে যেখানে প্রবেশ নিষেধ, যেখানে তিনি বিরাজ করেন — তেনার নাম কম্পিউটার। এরপর মেশিনের জটিল দেহে ঢুকিয়ে দিতে হবে সেই প্লাগবোর্ড। চল্লিশ এবং পঞ্চাশের দশকে, বিশেষ করে নানা ধরনের হিশেবের কাজ, সে ব্যবসার হিশেবই হোক বা বিশুদ্ধবিজ্ঞানের গণিতের, একধারসে হয়েছে এই প্লাগবোর্ড দিয়ে। এই ব্যবস্থাটা কিন্তু, ইলেকট্রনিক নয়, ইলেকট্রিকাল। কার্ডের গায়ে নানা তারের কানেকশন জুড়ে জুড়ে তৈরি হত প্লাগবোর্ড। বোর্ডের গায়ে থাকত সকেট, তার মধ্যে কেবলগুলো গুঁজে দেওয়া হত। এই প্লাগবোর্ডই ধারণ করত এখুনি চালানোর প্রোগ্রামটাকে। প্লাগবোর্ড গুঁজে দিত এবং ইষ্টনাম জপ করত প্রোগ্রামার। কম্পিউটারের মধ্যে ভরা হাজার বিশেক ভ্যাকুয়াম টিউব বা ভালভের একটাও যদি বিগড়োয় তাহলেই গেল। গোটা কাজটাই হত সরাসরি হিশেবের — রাশি রাশি সাইন, কস, ট্যান, বা লগারিদম, বা কোটি কোটি যোগ বিয়োগ। তার মানে, আমাদের তিন নম্বর দিনের আলোচনার সঙ্গে মিলিয়ে বলা যায়, শুধু ব্যাবেজের অ্যানালিটিকাল ইঞ্জিনে যা হত গিয়ার গুলি ডান্ডা দিয়ে, মানে মেকানিকাল উপায়ে, এখানে সেই কাজটা হচ্ছে লজিক রিলে আর ভালভ দিয়ে, মানে ইলেকট্রনিকালি। কিন্তু কম্পিউটার মননের স্তরে তেমন কোনো বদল তখনো আসেনি।

১৯৫০-এর গোড়ার দিকে এল ওই পাঞ্চড কার্ড, যার কথা আগেই বলেছি আমরা। কার্ডের গায়ে ফুটো করে করে কম্পিউটারকে কাজের নির্দেশ এবং সেই কাজের জন্যে প্রয়োজনীয় তথ্য দেওয়া। তিন নম্বর দিনে, যে কার্ড নিয়ে বেশ কিছু কথা বলেছি আমরা। পাঞ্চড কার্ড আসায় প্লাগবোর্ড অচল হয়ে গেল। কিন্তু এইটুকু বদলকে ছেড়ে দিলে মেশিনগুলো বা তাদের কাজের পদ্ধতিটা রয়ে গেল প্রায় একই।

৩।। দ্বিতীয় জেনারেশন কম্পিউটার — ১৯৫৫-৬৫

এবার ভ্যাকুয়াম টিউবের দিন শেষ। এল ট্রানজিস্টর। এবং তার সঙ্গে এল ব্যাচ সিস্টেম। মানে একসাথে তালিকাবদ্ধ অনেকগুলো কাজ পর পর করার ব্যবস্থা। পরে, আমাদের এই পাঠমালায় আবার আমরা দেখব, এবং আগেও উল্লেখ

করেছি, এই ব্যাচ সিস্টেম ব্যাপারটার মধ্যে কিন্তু প্রোগ্রামিং ক্রিয়ার বেশ কিছু চিন্তন উপাদান উপ্ত হয়ে আছে। পরে, শেল স্ক্রিপ্টিং-এর প্রসঙ্গে আমরা ফেরত আসব এই কথায়।

বাজারে ট্রানজিস্টর এসে গিয়ে পুরোটাকে একসঙ্গে অনেকটা বদলে দিল। কম্পিউটার ব্যাপারটাই আগের চেয়ে অনেক নির্ভরযোগ্য হয়ে উঠল — যাতে বানিয়ে লোককে বেচা যায়, এতদিন কম্পিউটারগুলো চলত অহর্নিশি নির্মাতার তত্ত্বাবধানে, এখন সেটা ক্রমে কমতে শুরু করল। এতদিনকার ওই কম্পিউটার পিছু এক সেট মানুষের মধ্যে এবার ভাগ তৈরি হল। কেউ কম্পিউটারের ডিজাইন তৈরি করে — ডিজাইনার, কেউ বানায় — বিল্ডার, কেউ চালায় — অপারেটর, কেউ প্রোগ্রাম লেখে — প্রোগ্রামার, কেউ দেখভাল করে — মেইনটেইনার। এতদিন অব্দি এদের প্রত্যেককেই প্রায় প্রত্যেকটা কাজই করতে হত।

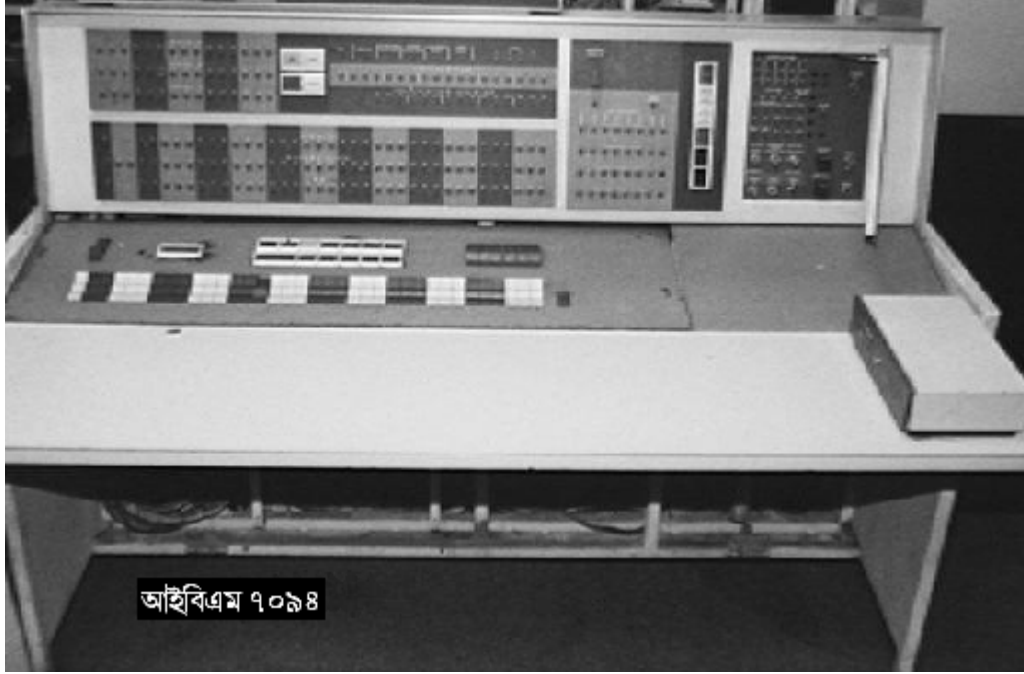
এই মেশিনগুলোর চালু নাম এল — মেইনফ্রেম। চাবি-লাগানো দরজার ওপারে বাতানুকূল কামরায় নির্দিষ্ট কিছু অপারেটর তাকে চালায়। আর সেই জগদ্দলের দাম যা তাতে সরকার বা বিশ্ববিদ্যালয় ছাড়া কারুর পক্ষে কেনা অসম্ভব। যে কোনো কাজ, সে একটা প্রোগ্রামের হোক, বা অনেকগুলো, প্রোগ্রামারকে প্রথমে সেটা কাগজে কলমে লিখে ফেলতে হবে, হয় অ্যাসেম্বলি কোডে, অথবা ফোর্ট্রান নামের কম্পিউটার ভাষায় (FORTRAN — **FOR**mula-**TRAN**slation)। বিজ্ঞানের কাজের জন্যে তৈরি কম্পিউটার ভাষা ফোর্ট্রান তদ্দিনে এসে গেছে। বানিয়েছেন জন ব্যাকাস, ১৯৫৪ থেকে ৫৮-য়। পরবর্তীকালের হাইলেভেল কম্পিউটার ভাষাগুলোর প্রচুর ধারণা এসেছে এই ফোর্ট্রান থেকে। বিজ্ঞান এবং ইঞ্জিনিয়ারিং-এর কাজে এখনো ব্যাপকভাবে ব্যবহার হয়।

এর পর, কাগজে-কলমে লেখা প্রোগ্রামটাকে নিয়ে যেতে হত কার্ড লেখার মেশিনে। কার্ড-লেখার মেশিন দিয়ে এই এই প্রোগ্রামটাকে পাঞ্চড কার্ডের গায়ে ফুটোর অক্ষরে লিখে দেওয়া হত। ফুটো-বিভূষিত সেই কার্ডমালা তারপর নিয়ে আসা হত সর্বসাধারণের প্রবেশ-অযোগ্য সেই কম্পিউটার কক্ষে, অপারেটরের দায়িত্বে। প্রোগ্রামারের তখন ছুটি। এখন থেকে দায়িত্ব অপারেটরের, আগের প্রোগ্রামের চলতে থাকা কাজ শেষ হলে পুরোনো কার্ডমালা বার করে অপারেটর নতুন প্রোগ্রামের নতুন সেট গুঁজে দেবে কম্পিউটারের পেটে। যদি প্রোগ্রামটা লেখা হয় ফোর্ট্রানে, তাহলে ফোর্ট্রান কম্পাইলারের নিজস্ব কার্ডমালাও যোগ করে দিতে হবে সঙ্গে, আগেই তো বললাম, আদেশ বোঝার ভাষা কম্পিউটারকে বোঝাতে গেলে তার জন্যে দিতে হয় সরলতর আদেশ, সেটাই কম্পাইলারের কাজ।

কম্পিউটারের বেশির ভাগ সময়টাই নষ্ট হত অপারেটারদের কার্ড ঢোকানো বার-করা, আর একটা প্রোগ্রামের পর আর একটা প্রোগ্রামের ফাইল নিয়ে আসার কাজে। আর নষ্ট করার পক্ষে কম্পিউটার-সময়ের দামটা ছিল বড্ড বেশি। তাই এল ব্যাচ-সিস্টেম। যাতে পর পর অনেকগুলো কাজ একসঙ্গে তুলে দেওয়া হত ম্যাগনেটিক টেপে। এই তুলে দেওয়ার কাজটা হত তুলনামূলক ভাবে অনেক সস্তা একটা কম্পিউটারে। এই ধরনের কম্পিউটারের একটা ভালো উদাহরণ আইবিএম ১৪০১। ১৪০১ মেশিনটা এই কাজের জন্যেই তৈরি। যাতে অজস্র কার্ড পরপর খুব তাড়াতাড়ি পড়ে ফেলতে পারে, এবং সেই সমস্ত কার্ডের যাবতীয় তথ্য এবং আদেশ এবং সেই আদেশ পালনের

জন্যে প্রয়োজনীয় সরলতর আদেশ — এই যাবতীয় তথ্য — মানে কমান্ড আর ডেটা এই দুই রকম ডেটা — ঝটাঝট লিখে ফেলতে পারে ম্যাগনেটিক টেপে। কিন্তু অন্য ধরনের কাজে, গাণিতিক হিশেবের কাজে অন্য উন্নততর কম্পিউটারের মত দক্ষ নয় ১৪০১।

তারপর, ঠিক যেমন একটা ক্যাসেট গানে ভরে গেলে রেকর্ড হওয়া ক্যাসেটটা আমরা ঘুরিয়ে শুরুতে নিয়ে যাই বাজানোর জন্যে, একটা নির্দিষ্ট সময় ধরে ১৪০১-গোছের কম্পিউটার দিয়ে যথেষ্ট পরিমাণ কমান্ড আর ডেটা টেপে জমা হয়ে গেলে, টেপটা রিওয়াইন্ড করা হত, মানে ঘুরিয়ে গোড়ায় নিয়ে যাওয়া হত, যাতে সেটা থেকে তথ্য এবং আদেশ পড়ার সময় শুরু থেকে শুরু করা যায়। আর ভর্তি হয়ে যাওয়া ইনপুট টেপটা নিয়ে যাওয়া হত আসল কম্পিউটারে, যে কম্পিউটার অঙ্ক কষার কাজটা, মানে আসল কাজটা অনেক বেশি গতিতে করতে পারে। এই পরের কম্পিউটারটা অনেক বেশি দামী, যেমন আইবিএম ৭০৯৪। ওই তথ্য আর আদেশ ভরা টেপটা এই ৭০৯৪ কম্পিউটারে ঢোকানোর আগে অপারেটর এবার ঢোকাবে একটা বিশেষ প্রোগ্রাম, আজকের অপারেটিং সিস্টেমের কোনো আদি আদি পূর্বপুরুষ, যিনি কম্পিউটারকে বলে দেবেন ওই টেপে লেখা পরপর কাজগুলো কম্পিউটার কী ভাবে করে যাবে। এই আদি অপারেটিং সিস্টেমই এবার পড়তে শুরু করবে টেপটা এবং টেপে লেখা আদেশ মোতাবেক টেপে লেখা তথ্য চটকানোর কাজটা শুরু করে দেবে।

আইবিএম ৭০৯৪

৭০৯৪ আবার যে কাজগুলো করবে তার ফলাফলগুলো লিখে রাখবে অন্য একটা টেপে, তার নাম আউটপুট টেপ। যেই একটা করে কাজ শেষ হবে, অপারেটিং সিস্টেম কাজের লাইনে বসে থাকা পরবর্তী কাজটা ঢুকিয়ে দেবে মেশিনে। কাজের পুরো ব্যাচটা হয়ে গেলে অপারেটর এবার সেই প্রাথমিক ইনপুট টেপ আর ৭০৯৪-এর তৈরি এই আউটপুট টেপটা দিয়ে দেবে প্রোগ্রামটা যে দিয়েছিল তার কাছে। প্রোগ্রামার আবার সেই টেপ নিয়ে ফেরত যাবে ১৪০১-এর কাছে। সে ওই টেপ থেকে ফলাফলটার প্রিন্টআউট বার করে দেবে তার সিস্টেমের প্রিন্টার দিয়ে।

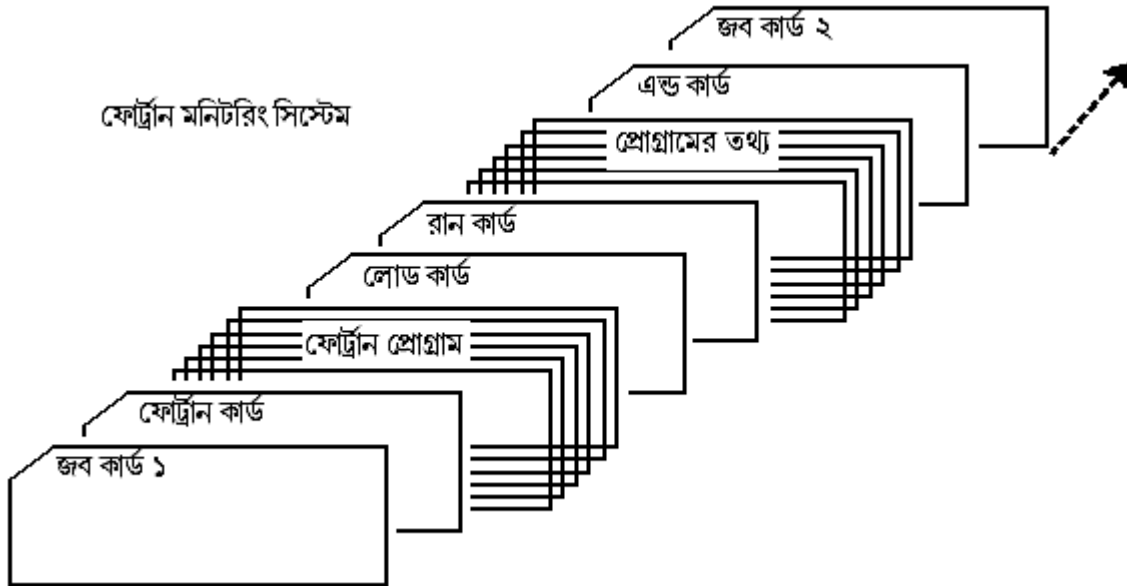
আজ আমরা লিখছি, বা প্রোগ্রাম করছি, চালাচ্ছি, কমান্ড দিতে না দিতেই কাজ হয়ে যাচ্ছে, কাজ শেষ করা মাত্রই ফাইলটা প্রিন্টার পোর্ট এলপিটি-ওয়ানে (এই এলপিটি ওয়ান বা LPT1 — **L**ine-**P**rint-**T**erminal-numbered-1, লাইন- প্রিন্ট-টার্মিনালও সেই অর্থে একটা লুপ্ত মেটাফর, এখন আর এই ভাবে প্রিন্টই হয়না, নামটা চলে আসছে সেই তখন থেকে যখন এভাবে প্রিন্ট হত) কপি করার মানে প্রিন্ট করার আদেশ দিচ্ছি, বা গুইতে মানে ছবি আর মাউস দিয়ে কাজ করলে কেডিট বা ওওও-র (OOo — **O**pen**O**ffice.**o**rg) রাইটারের মেনুবারে প্রিন্টারের ছবি দেওয়া আইকনে ক্লিক করছি, খ্যাচখ্যাচ করে প্রিন্ট বেরিয়ে আসছে, আমাদের পক্ষে আজ চেষ্টা করলেও আন্দাজ করা শক্ত ওই সময়ের কম্পিউটার চালানোর সামগ্রিক ঝঞ্ঝাটটা। একটা জিনিষ খেয়াল করুন, শূন্য নম্বর দিনে আমরা

কম্পিউটারের যে কাজভিত্তিক ছকটা দিয়েছিলাম, সেটা কিন্তু এখানে সত্যি হচ্ছে একটা কম্পিউটারে নয়, আইবিএম ১৪০১ আর আইবিএম ৭০৯৪ এই দুটো মেশিনকে মিলিয়ে।

৪।। ফোর্ট্রান মনিটরিং সিস্টেম — অপারেটিং সিস্টেমের আদিপুরুষ

এই আদিতম অপারেটিং সিস্টেম তার বশংবদ ১৪০১ আর ৭০৯৪-কে দিয়ে কাজটা যে করাল, সেই কাজটার মধ্যে কিন্তু একটা খুব সুস্পষ্ট কাঠামো আছে। ১৪০১-এ সরাসরি হয়ত অপারেটিং সিস্টেম-এর কোনো অস্তিত্ব নেই, কিন্তু ৭০৯৪-এ যে কাজটা শেষ হচ্ছে, অপারেটিং সিস্টেম-এর তত্বাবধানে, তার শুরুটা হয়েছিল তো ১৪০১-এ তথ্য তোলা থেকে। সেই অর্থে তাই গোটা কাজটা ঘটছে দুটো মেশিনকে মিলিয়েই। আমাদের মধ্যমগ্রাম জিএলটির অন্তত দুজনকে দেখেছি এটা নিয়ে জিগেশ করতে — আদেশ আর তথ্য — দুটোই তো তথ্য। সেদুটোকে আলাদা করা হচ্ছে কী করে? আমরা এখানে গন্ডগোল পাকাচ্ছি কাজ শব্দটা দিয়ে। কাজটা একটা কাজ, আবার কাজটা পালন করাটাও কাজ। আবার, ইয়ে, অকাজও তো কাজ, আমাকে এত বছর ধরে দেখার পর মা এই সুস্পষ্ট সিদ্ধান্তে পৌঁছেছে — আমি 'কাজ' শব্দটা উচ্চারণ করলেই মুখটা কিরকম বেগুনভাজার মত (কার্টসি প্যালা) করে ফেলে।

তার চেয়ে ১৪০১ টেপে করে ৭০৯৪-এর কাছে যা পাঠাল তাকে ডাকা যাক 'জব' বলে। এবার ৭০৯৪-এর মোট কাজটাকে যদি কাজ বলে ডাকি, যা চলছে ওই আদি অপারেটিং সিস্টেমের অতন্দ্র তত্ত্বাবধানে, তাহলে সেই 'কাজ' কিন্তু তৈরি 'জব ১', 'জব ২' . . . ইত্যাদি পরপর টুকরোগুলোকে জুড়ে, কিন্তু শুধু ওই টুকরোগুলোই নেই, আরো কিছু আছে সেখানে। এখানে এই আলোচনাটা পরে অপারেটিং সিস্টেমের কাঠামো বুঝতে আমাদের খুবই কাজে লাগবে, বারবার, খেয়াল করে পড়ুন।

ধরুন ওই আদি অপারেটিং সিস্টেমের পাহারায় পরপর তার জবগুলোকে করে যাচ্ছে ৭০৯৪। এবার সে একটা জবকে জব বলে চিনছে কী করে? প্রথমে নিশ্চয়ই জব বলে চেনানোর একটা রকম থাকবে। রকম আর কী, সবই তো কার্ড দিয়ে হচ্ছে। তার মানে প্রথমে একটা কার্ড থাকবে যা বলে দেবে এবার আসছে এই কাজ। যে কার্ডে কী কাজ, কত বড়, কত তারিখ কী সময়ে কে দিয়েছে। এগুলো থাকত কম্পিউটার মালিক যাতে তার নিজের হিশেব রাখতে পারে — কে কতটা কাজ করাচ্ছে তার মেশিনকে দিয়ে। এবং, আগেই বলেছি, সেইসময় একটা মেশিনের যা দাম, একটা সংস্থার নিজের একটা কম্পিউটার খুব কমই থাকত। বেশিরভাগ ছোট বা মাঝারি মাপের কোম্পানি কোনো একটা বড় সংস্থার কম্পিউটার থেকে এই কাজগুলো করিয়ে নিত।

যাই হোক, কাজের খুঁটিনাটি লেখা এই কার্ডটার আমরা নাম দিলাম জব-কার্ড। এবার এই জব-কার্ডে দেওয়া কাজটা চালানোর জন্যে ফোর্ট্রান প্রোগ্রামটা লাগবে। তার মানে আবার একটা কার্ড, ফোর্ট্রান প্রোগ্রামটার নামপত্র, ধরুন এর নাম দিলাম ফোর্ট্রান-কার্ড। এইটা দেখে অপারেটিং সিস্টেম বুঝতে পারবে যে এই প্রোগ্রামটা চালাতে হবে ফোর্ট্রান

নামক কম্পাইলার দিয়ে। ফোর্ট্রান কম্পাইলার মানে ফোর্ট্রান ভাষায় লেখা প্রোগ্রাম চালাতে গেলে কম্পিউটারকে যে আদেশগুলো দিয়ে রাখতে হয়, আমরা আগেই বলেছি। অপারেটিং সিস্টেম এবার টেপ থেকে ফোর্ট্রান কম্পাইলারের আদেশগুলো পড়ে নেবে। টেপ থেকে এই পড়াটা হয়ে গেলে একটা আদেশ দিতে হবে এবার প্রোগ্রামটা চালানোর। কিন্তু, চালানোর আগে দুটো স্তরে কাজ সারতে হবে। প্রথম স্তরে মানুষবোধ্য ফোর্ট্রান ভাষায় লেখা কোডটাকে তুলে নিতে হবে কম্পিউটারে। তারপরে, দ্বিতীয় স্তরে সেটাকে কম্পাইল করতে হবে। মানে প্রোগ্রামারের হাইলেভেল ভাষায় লেখা বিশেষ আদেশমালাকে কম্পিউটারের কাছে যন্ত্রের ভাষায় বোধ্য করে তুলতে হবে। যেই কম্পিউটারবোধ্য করা হল, কম্পাইল করার কাজ শেষ হল। যন্ত্রে চালানোর মত প্রোগ্রাম প্রস্তুত হয়েছে। এবার চালানো যায় মেশিন।

কম্পাইল করা যন্ত্রবোধ্য প্রোগ্রামটাকে এবার মেশিনের চালু মেমরিতে তুলে নিতে হবে। মানে, আর একটা কার্ড। ধরুন সেটার নাম লোড-কার্ড। এবার আসবে সরাসরি কম্পাইলড প্রোগ্রামটা চালানোর আদেশ। ধরুন এই কার্ডটার নাম রান-কার্ড। এবার রান তো করবে মেশিন, কিন্তু কার উপর রান করবে? তার মানে, এই প্রোগ্রাম দিয়ে চটকানোর জন্যে প্রয়োজনীয় তথ্য। ধরুন একটা বড় অঙ্ক কষা হচ্ছে, কষতে হবে যে ফরমুলা দিয়ে সেটা তো প্রোগ্রাম করে আপনি ঢুকিয়ে দিয়েছেন, এবার কষতে হবে যে সংখ্যাগুলো দিয়ে সেগুলো ঢোকাতে হবে। শুধু অঙ্ক কষা হলে এই তথ্য মানে শুধু সংখ্যা বা নিউমেরিক তথ্য, অন্য কাজ হলে টেক্সট এবং অঙ্ক এই দুরকম মানে আলফানিউমেরিক তথ্য। তথ্যের এই রকমগুলো শূন্য নম্বর দিন থেকে দেখে নিন। অপারেটিং সিস্টেম এবার সেই তথ্যগুলো পড়বে টেপ থেকে। প্রোগ্রাম চলবে। এরপর, সব শেষ হলে, শেষ হওয়ার ঘোষণা, মানে এন্ড-কার্ড। এরপর আবার ঘুরে আসবে আর একটা কাজ, জব-কার্ড ২। আবার এই রকম একটা সিরিজ চলবে, সেটা হয়তো আবার ফোর্ট্রানে হবে না, হবে অ্যাসেম্বলি ভাষায়। তার মানে তার জন্যে আবার আলাদা রকমে কাজ করতে হবে অপারেটিং সিস্টেমকে।

দ্বিতীয় জেনারেশনের এই গাবদা কম্পিউটারগুলো মূলত ব্যবহার হত বড় বড় বিজ্ঞান গবেষণার আর ইঞ্জিনিয়ারিং-এর কাজে। মানে, মূলত বিশালাকার সব গাণিতিক হিশেব। যেমন পদার্থবিদ্যার বা আর্কিটেকচারের গাদা গাদা পার্শিয়াল ডিফারেনশিয়াল ইকুয়েশন সমাধান। এর বেশির ভাগ কাজটাই হত ফোর্ট্রানে বা অ্যাসেম্বলি ল্যাংগুয়েজে। এবং অপারেটিং সিস্টেমের বংশলতিকার ওই আদিপুরুষটি বেশিরভাগ ক্ষেত্রেই ছিল এফএমএস (FMS — **F**ortran-**M**onitoring-**S**ystem) বা আইবিএম-এর আইবিসিস। আমরা এখানে ফোর্ট্রান-মনিটরিং-সিস্টেমের একটা কাজের ছক দেখালাম।

৫।। তৃতীয় জেনারেশন কম্পিউটার — ১৯৬৫-১৯৮০

ততদিনে কম্পিউটার তৈরির দুটো টাইপ প্রায় আলাদা হয়ে এসেছে। একটা মেশিন বড় আকারের, বড় বড় আঙ্কিক হিশেব করার — মূল মনোযোগটা এখানে ওয়ার্ড বা শব্দের উপর। ওয়ার্ড মানে স্মৃতি বা মেমরির একক। এরা বড় বড় হিশেব এবং হিশেবনির্ভর কাজ করবে। ব্যবহার হবে বিজ্ঞান আর ইঞ্জিনিয়ারিং-এর কাজে। একটু আগে যেমন আইবিএম ৭০৯৪ দেখলাম আমরা। আর অন্য কম্পিউটারটা হল অপেক্ষাকৃত ছোট আকারের কমার্শিয়াল কম্পিউটার — মূলত ব্যবহার হবে ইনপুট তথ্যকে এবং আদেশকে টেপে ঢোকানো, আর আউটপুট তথ্যকে টেপ থেকে বার করে আনার কাজে। যেমন, একটু আগের আইবিএম ১৪০১। মূল মনোযোগটা এখানে ওয়ার্ড বা শব্দ নয়, ক্যারেকটার বা অক্ষর। শব্দ বা ধারণা নয়, শুধু অজস্র অক্ষরের পর অক্ষরের সমাহারে রাশি রাশি তথ্য। মূলত টেপে তথ্য তোলা এবং টেপ থেকে তথ্য পড়ার কাজে, এবং টেপে করে আসা তথ্য প্রিন্টারে প্রিন্ট নেওয়ার কাজে ব্যাপকভাবে এদের ব্যবহার করত অনেক ব্যাংক এবং বীমা কোম্পানি।

প্রযুক্তিবদলের সঙ্গে সঙ্গে দামও কিছু কমছিল, আর কম্পিউটার দিয়ে কাজ করানোর প্রয়োজনও বাড়ছিল। সরকার এবং বিশ্ববিদ্যালয় এবং মূলত ব্যবসা-প্রতিষ্ঠানগুলো আরো বেশি বেশি করে মেশিন কিনছিল। কিন্তু তাদের প্রয়োজনগুলো তো আর কম্পিউটারের ধরনের সঙ্গে মিলিয়ে তৈরি হয়নি। প্রায়ই তাদের, একইসঙ্গে, দুরকম মেশিনের দুরকম কাজেরই দরকার পড়ছিল। বা, কিছুদিন ধরে চলতে থাকা এক ধরনের প্রয়োজন প্রায়ই বদলে যাচ্ছিল অন্য ধরনের প্রয়োজনে। তাই, এই দু টাইপের কাজ একই মেশিনে করতে পারার চাহিদা ক্রমে বাড়ছিল। আইবিএম আনল সিস্টেম/৩৬০ বলে একটা যন্ত্র। যা আগের বড় কম্পিউটারগুলোর সমান ক্ষমতার, এবং আকারে

অনেক ছোট। আর সবচেয়ে বড় কথা, এরা শুধু হার্ডওয়ার স্তরের অ্যাসেম্বলি কোড দিয়েই চলে না, সফটওয়ারও বোঝে। আর যেহেতু এই সিরিজের সবগুলো মেশিনেরই গঠন এক, একটা কম্পিউটারের প্রোগ্রাম নির্দ্বিধায় অন্য কম্পিউটারে চালানো যায়। ওয়ার্ডনির্ভর বিজ্ঞানের কাজ আর ক্যারেকটারনির্ভর ব্যবসায়িক তথ্যের কাজ — দুই-ই করতে পারে এই সিস্টেম/৩৬০।

এই আইবিএম সিস্টেম/৩৬০ মেইনফ্রেম সিস্টেমে ব্যবহারের জন্যে এসেছিল জেসিএল (JCL — **J**ob-**C**ontrol-**L**anguage), একটা কমান্ড ল্যাংগুয়েজ যা দিয়ে কোনো প্রয়োগমূলক অ্যাপ্লিকেশন প্রোগ্রাম চালু করা যেত এবং ওই প্রোগ্রামের আকার, কত সময় ধরে চলবে, এবং চলতে গিয়ে প্রোগ্রামটা কী কী ফাইল ব্যবহার করবে — এইসবই ঠিক করে দেওয়া যেত। এই কমান্ড ল্যাংগুয়েজ হল কমান্ড ইন্টারপ্রিটারের কাজের একটা অংশ। ব্যাশ শেলের প্রসঙ্গে কমান্ড ইন্টারপ্রিটারের কথা পেয়েছি আমরা, আরো পাব।

সিস্টেম/৩৬০ মেশিনেই এল আইসি বা ইন্টিগ্রেটেড সার্কিট। দ্বিতীয় জেনারেশন কম্পিউটারগুলোর সঙ্গে দাম এবং ক্ষমতার লেভেলে সবচেয়ে বড় তফাত গড়ে দিল এই আইসি। জনপ্রিয় এই সিস্টেম/৩৬০ অনেক বিক্রি করেছে আইবিএম, পরে এই সিরিজে অন্য মডেলও এনেছে। যারা পরস্পর কম্পেটিবল বা স্থানান্তরযোগ্য, মানে এমন কম্পিউটার যাদের একের কাজ বা তথ্য বা প্রোগ্রাম অন্যের কাছে গ্রহণযোগ্য। কম্পেটিবল হওয়ার ধারণাটাও কম্পিউটার নির্মাতাদের কাছে জনপ্রিয় করেছে আইবিএম। সিস্টেম/৩৬০-এর কিছু উত্তরপুরুষ আজো ব্যবহৃত হচ্ছে, বিশাল বিশাল ডেটাবেস নিয়ন্ত্রণের কাজে, যেমন টিকিট রিজার্ভেশন বা ওই গোছের, বা ইন্টারনেটের সার্ভার হিশেবেও। পরবর্তী বছরগুলোয় আইবিএম এই সিস্টেম/৩৬০ সিরিজে আরো অনেক মডেল এনেছিল — ৩৭০, ৪৩০০, ৩০৮০, ৩০৯০ ইত্যাদি।

আইবিএম
সিস্টেম/৩৬০

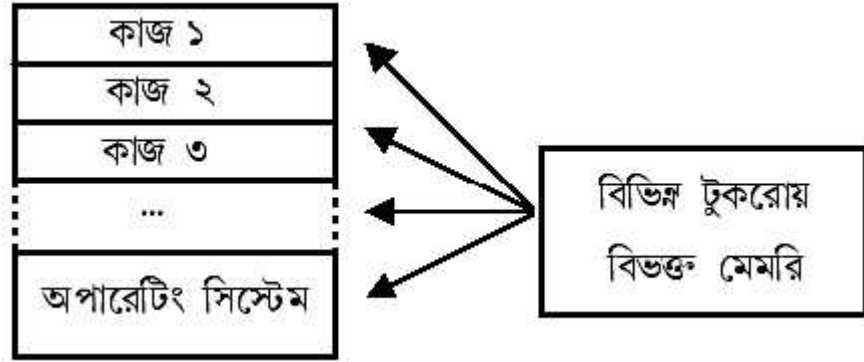
কিন্তু এই পরস্পর কম্পেটিবল কম্পিউটার পরিবারের ধারণাটায় সুবিধে যেমন অসুবিধেও আছে। নীতিটা এই যে, অপারেটিং সিস্টেম সমেত প্রতিটি সফটওয়ার এই পরিবারের প্রতিটি মেশিনে চলবে। তার মধ্যে কার্ড ফোটানোর দুধেভাতে কম্পিউটারও আছে, আছে আবহাওয়া পূর্বাভাসের মৈনাক মেশিনও। আর কুচো থেকে মৈনাকে তফাত শুধু আকারের নয়, তফাত তাদের কাজে, কাজে ব্যবহৃত তথ্যে, এবং সবচেয়ে বড় তফাত পার্শ্বীয় যন্ত্রপাতিতে মানে পেরিফেরাল ডিভাইসে। তার মানে একটা বিরাট সংখ্যক পেরিফেরালের প্রত্যেকটার সঙ্গে ব্যবহার করতে পারার ক্ষমতা এই পরিবারের প্রত্যেকটা কম্পিউটারেই ভরে দিতে হবে। তার মানে, দু নম্বর দিন থেকে মনে করুন — নানা ধরনের ড্রাইভার। ব্যবসা জগতের পেরিফেরাল থেকে বিজ্ঞান জগতের পেরিফেরাল।

এই আলপিন টু এলিফ্যান্ট প্রতিটি মালের প্রতিটি চাহিদার সঙ্গেই চলনসই কোনো একটিমাত্র সফটওয়ার তৈরি করা আইবিএম কেন, কারো পক্ষেই সম্ভব ছিল না। এর ফলে যা হবার তাই হল। একটা বেধড়ক বদখত রকমের বড় সাইজের জটিল বহুমুখী অপারেটিং সিস্টেম। দ্বিতীয় জেনারেশন কম্পিউটারের ওই এফএমএস বা আইবিএমসিসের অন্তত তিনগুণ সাইজের। কোটি কোটি লাইনের অ্যাসেম্বলি কোড — যার পরতে পরতে অর্বুদ অর্বুদ বাগ। বাগ মানে তো বলেছি, প্রোগ্রামের ভুল, মানে খিঁচ, মানে মেশিন ঝুলে গেল, যা করার কথা তা না-করে আর সমস্ত কিছু

করারই অন্য নাম বাগ। এই পুরোনো দেড় হাজার বাগ শায়েস্তা করতে লেখা হল নতুন করে সোয়া দুলক্ষ লাইন অ্যাসেম্বলি কোড, তার লাইনে লাইনে জন্ম নিল আরো পৌনে দুহাজার বাগ। পোকামাকড় সংক্রান্ত এই বীভৎসাবিলাস সত্ত্বেও তৃতীয় জেনারেশন কম্পিউটার, সিস্টেম/৩৬০ এবং তাদের ওই ঘটোৎকচ অপারেটিং সিস্টেম পয়দা করে গেল একটা চমৎকার নতুন ধারণা — মাল্টিপ্রোগ্রামিং।

৬।। মাল্টিপ্রোগ্রামিং

সিস্টেম/৩৬০ এবং ওই জাতীয় নতুন কাঠামোগুলোয় আইসি বা ইন্টিগ্রেটেড সার্কিট ছাড়া আর একটা যে অভিনব বিষয় আমরা পেলাম সেটা হল মাল্টিপ্রোগ্রামিং। এই সিস্টেমগুলো এদের ভিতরকার ওই অযুত পোকামাকড় নিয়েই খদ্দেরদের চাহিদা বেশ ভালই মেটাচ্ছিল। এই মেটানোর প্রক্রিয়ায় যে সমস্ত নতুন নতুন ধারণাকে বাজারে এনেছিল তৃতীয় জেনারেশন, তার মধ্যে সবচেয়ে গুরুত্বপূর্ণ এই মাল্টিপ্রোগ্রামিং।

এর আগেকার বড় বড় সিস্টেমগুলোয় চলমান একটা কাজ যখন একটু থামত, নতুন টেপ গোঁজার ব্রেকে, বা, কোনো মহাভারত মাপের ইনপুট-আউটপুট জব শেষ হওয়ার অবসরে, হয়তো বহু পাতার একটা বড় প্রিন্ট চলছে, সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট একদম চুপচাপ বসে থাকত — যতক্ষণ না টেপটা গোঁজা হয় বা প্রিন্টটা শেষ হয়, আবার কাজ শুরু করার সুযোগ আসে। বহু বহু লাইনব্যাপী কোনো খেপচুরিয়াস বৈজ্ঞানিক হিশেবের খুব সিপিইউ-নিবিড় কোনো কাজ চলাকালীন এই ধরনের আইও ইন্টারমিশন হয়ত কমই হত, মানে সিপিইউ-র কাজের সময়ের তুলনায় বসে থাকার সময়ের হার খুব কমই থাকত। কিন্তু অন্য অন্য কাজে, কোনো বাণিজ্যিক ডেটা প্রসেসিং বা ওই ধরনের কিছু, এই ইনপুট-আউটপুট অপেক্ষার প্রহর বেড়ে বেড়ে পৌঁছে যেত আশি বা নব্বই শতাংশে। সিপিইউ-র সক্রিয় থাকার সময়ের হার কখনো কখনো নেমে গিয়ে মোট সময়ের দশ শতাংশেও দাঁড়াত। তার মানে, অন থাকার মোট সময়ের দশ শতাংশ মাত্র কম্পিউটারের মূল শক্তিটা ব্যবহৃত হচ্ছে। এ তো ঠিক না। এত দাম দিয়ে কেনা মেশিন এতক্ষণ কেন বসে থাকবে?

এর সমাধান আকারে এল একটা নতুন পদ্ধতি। স্মৃতি বা মেমরিটাকে পিসপিস করে ফেলা ক্ষুদ্রতর কিছু টুকরোয়। দুই নম্বর দিনে আমাদের মাল্টিপ্লেক্সিং নিয়ে আলোচনাটা মনে করুন। স্মৃতি বা মেমরির এই প্রতিটি পিসের হাতে এখন এক একটা ছোট ছোট কাজ ধরিয়ে দেওয়া হবে। কাজ নম্বর ১ যখন আইও ব্রেক মেটার অপেক্ষা করছে জিভ পুড়িয়ে ফেলা গরম চায়ের কাপে চুমুক দিতে দিতে, কাজ নম্বর ২ সঙ্গোপনে সিপিইউর সঙ্গে ডাবল-টাইম ডেটিং শুরু করে দিয়েছে — ইংরিজিতে যাকে বলে পরকিয়া। এবং এই গোটা সময়টা ধরে এই শতধা বিদীর্ণ মেমরির একটা বড় খণ্ড যত্নআত্তি করে যাচ্ছে প্রভুর প্রভু অপারেটিং সিস্টেমের। কখন সে কোন প্রসেসকে কোন আদেশ পাঠাবে বা কোন প্রসেসের থেকে আসা কোন আবেদনকে মঞ্জুর করবে — এই তথ্য এদিক থেকে ওদিক এবং ওদিক থেকে এদিক বহন প্রতিবহন করবার জন্যে। এই পুরো ব্যাপারটার বিউটি এইখানে যে এতে করে মহার্ঘ সিপিইউ-সুন্দরীকে প্রায় গোটা সময়টাই প্রায় একশো শতাংশতেই ব্যবহার করা যাচ্ছে। শুধু, সিপিইউকে মাল্টিপ্রোগ্রাম পদ্ধতিতে এই বহুগামিনী করে তোলার শর্ত একটাই। তখনকার সময়ের তুলনায় একটু আলাদা রকমের একটা হার্ডওয়ার যা স্মৃতির প্রতিটি খণ্ডকেই তার নিজস্ব একটা মহল বানিয়ে দেবে, এবং তার সঠিক প্রতিরক্ষার ব্যবস্থা করবে যাতে সে অন্যদের সঙ্গে ঘেঁটে না-যায়। স্মৃতির একটা টুকরোয় ভরে রাখা তথ্য যাতে অন্য টুকরোর মধ্যে উপচে না যায়। কোনো একটা

জব যাতে অন্য কারো ঘরে ছিঁচকেমি না করতে পারে। এই ছিঁচকেমির টেকনিকাল নাম নাম সেগমেন্টেশন ফল্ট ইত্যাদি — যার কথায় আমরা পরে আসব। ইন্টিগ্রেটেড সার্কিটের উদ্ভাবন এই তৃতীয় জেনারেশনেই, আগেই বলেছি, তার সঙ্গে সঙ্গে এই নতুন ধরনের হার্ডওয়ারের চাহিদাও মেটাতে হয়েছিল ওই সময়ের ইলেকট্রনিক ইন্ডাস্ট্রিকে।

তৃতীয় জেনারেশন কম্পিউটারের আর একটা বড় অবদান ছিল পাঞ্চড কার্ড পড়ার অযথা সময় নষ্টটাকে বন্ধ করা। যেই মুহূর্তে কম্পিউটার কক্ষে এসে ফুটোময় কার্ডমালা আবির্ভূত হল, সেই মুহূর্তেই তাদের পড়ে ফেলার ব্যবস্থা। এতে একটা কাজ শেষ করেই মেশিন পরের কাজে হাত দিতে পারে। সিপিইউ যখন একটা কাজ করায় রত, সেই সময়টা জুড়েই পরের কাজের প্রয়োজনীয় কার্ডমালা পড়ে রাখা হচ্ছে, মাল্টিপ্রোগ্রামিং ব্যবহার করে। এবার, চালু কাজটা নামানো মাত্রই সিপিইউর গরম তাওয়ায় পরের কাজ চড়িয়ে দেওয়া হচ্ছে। এর অন্য নাম স্পুলিং। একসঙ্গে অনেকগুলো পেরিফেরালকে ব্যস্ত রাখার ব্যবস্থা (SPOOL — **S**imultaneous-**P**eripheral-**O**peration-**O**n-**L**ine)। স্পুল বাজারে ল্যান্ড করার সঙ্গে সঙ্গে উঠে গেল টেপ কাঁধে অপারেটরদের ওই অন্তহীন হাঁটাহাঁটি।

মাল্টিপল প্রোগ্রামিং, স্পুল ইত্যাদির দৌলতে অনেক বড় বড় বৈজ্ঞানিক হিশেবনিকেশ এবার করা যাচ্ছিল। তাও, এর পরেও, তৃতীয় জেনারেশন-ও শেষ অব্দি এক ধরনের উন্নততর ব্যাচ সিস্টেম। একদিকে ঝামেলা যেমন কমিয়েছিল, অন্যদিকে, টেপ জমা দেওয়া আর তার ফলাফল পাওয়ার মধ্যে বেশ কয়েক ঘন্টার ব্যবধান তৈরি হয়ে গেছিল, যা আগে ছিলনা। প্রোগ্রামে একটা ভুল সেমিকোলন, কম্পাইলেশন খতম, কাজ বন্ধ। কিন্তু ততক্ষণে একটা গোটা দিন কাবার। সেমিকোলন শুধরে আবার পরের দিন। অনেক প্রোগ্রামচিই তাই চাইছিলেন সম্পূর্ণ তার নিজের করে কম্পিউটারের সঙ্গে একান্তে কিছুক্ষণ। এর থেকে ক্রমে গজিয়ে উঠছিল টাইম-শেয়ারিং-এর চাহিদা — আর একটু আলাদা অবয়বের মাল্টিপ্রোগ্রামিং। যখন মূল সিপিইউ একই সঙ্গে একাধিক টার্মিনালে যুক্ত থাকবে। এবং এক একটা টার্মিনাল থেকে আসবে এক এক সেট কাজ। এই সমস্ত চাহিদাগুলোই বাজারে পৌঁছে তৈরি করছিল চতুর্থ জেনারেশন কম্পিউটারের সম্ভাবনা।

৭।। সময়ভাগের ব্যবস্থা — সিটিএসএস

তৃতীয় জেনারেশন কম্পিউটার ব্যবহার করতে করতে নতুন ধরনের চাহিদা তৈরি হচ্ছিল, মূলত, প্রোগ্রাম লিখে ফেলার পর থেকে প্রোগ্রাম চালিয়ে হাতে ফলাফল পাওয়ার মধ্যে এই প্রতীক্ষার প্রহরটাকে কমিয়ে আনার জন্যে, একান্তে শুধু আমি আর কম্পিউটার — প্রোগ্রামলিখিয়েদের এই বাসনা থেকে। তাই তৃতীয় জেনারেশনেই একটু একটু করে আসছিল টাইম-শেয়ারিং বা সময়ভাগ। তৃতীয় জেনারশন চলতে চলতেই, মাল্টি-প্রোগ্রামিং পদ্ধতিটা একটু একটু করে অন্য চেহারা নিচ্ছিল। এই প্রক্রিয়ায় কাজ ১ কাজ ২ কাজ ৩ ... ইত্যাদিতে গোটা রসদটাকে বরাদ্দ করে ভেঙে ভেঙে দেওয়া হচ্ছিল ঠিকই, কিন্তু সেই কাজগুলো সবই কম্পিউটারে গোঁজা হচ্ছিল একই টেলিটার্মিনাল থেকে, গ্নু-লিনাক্স-এ যাকে আমরা সংক্ষেপে 'tty' বলে ডাকি। এই টেলিটার্মিনাল শব্দটা আর একটা লুপ্ত মেটাফর। সেই সময় সেই প্রকৌশল সবই বদলে গেছে, শুধু বদলে যাওয়া ভাষায় তার পদচিহ্ন রেখে গেছে — আমরা আজো ব্যবহার করে চলেছি, নিজের পিসিতে লাগানো কনসোলকেও ডাকছি টেলিটার্মিনাল বলে, টাকা হিশেবে কড়ির ব্যবহার ভুলে গিয়েও যেমন তেল মাখতে গেলেই কড়ি ফেলি। যাকগে, এই জাতের সময়ভাগ ব্যবস্থাটাকে একবার ভাবার চেষ্টা করুন। কুড়ি জন প্রোগ্রামচি সমভিব্যাহারে যুক্ত সিপিইউ-র সঙ্গে, কুড়ি পিস টেলিটার্মিনালে তারা বসে আছে, তাদের কাজ পাঠাচ্ছে সিপিইউর কাছে, সিপিইউ তাদের যথারীতি ঠেলে দিচ্ছে বিভিন্ন ধরনের রসদের বরাদ্দ কামরায়, স্মৃতির হোক, হার্ডডিস্কের, বা এমনকি আইও বা ইনপুট-আউটপুটের লাইনে। দু নম্বর দিনে আমাদের আলোচনাটা মনে করুন, যেখানে আমরা স্পেস এবং টাইম, ভূমি এবং কাল, এই দুই রকমে মাল্টিপ্লেক্সিং নিয়ে কথা বলেছিলাম। কখনো পরপর খদ্দের আসছে লাইন বেয়ে, ঘুরে ঘুরে, ডিপার্টমেন্টাল স্টোরে যেমন হয় — কালগত মাল্টিপ্লেক্সিং। আর কখনো, হোটেলের রুমের মত, এক এক জন খদ্দেরকে এক এক সঙ্গে ছেড়ে দিচ্ছি এক একটা আস্ত ঘর, প্রথম জন ঘর ছেড়ে দিলে পরের জন ঢুকছে — ভূমিগত মাল্টিপ্লেক্সিং।

এবার, এটা স্ট্যাটিস্টিক্সের একটা সোনালি সূত্র যে, কুড়িখানা লোক যদি একই সঙ্গে বসে থাকে তাদের কাজ নিয়ে, তাদের মধ্যে সতেরোজন অবধারিত বসে বসে মস্তি মারছে, হয়তো গেমস খেলছে, হয়তো কফি খাচ্ছে, বা নিট আড্ডা দিচ্ছে পাশের প্রোগ্রামচির সঙ্গে। তার মানে একটা প্রদত্ত মুহূর্তে সিপিইউ-কে আদতে ডিল করতে হচ্ছে

তিনপিস কাজ। আর প্রোগ্রামচিদের সবাই যে নতুন প্রোগ্রাম কম্পাইল করছে তা নয়, অনেকেই পোকা বাছছে পুরোনো প্রোগ্রামের। ডিবাগিং-এর কমান্ডগুলো সচরাচর হয় খুব ছোট ছোট। কম্পিউটারের সিপিইউ খুব দ্রুত এদের চাহিদা মিটিয়ে দিয়ে অন্য কাজে ফিরে যেতে পারে। আর এর সঙ্গে, ব্যাকগ্রাউন্ডে, অন্য কাজগুলোর ফাঁকে দম ফেলার ফুরসতের সময়খণ্ডে সিপিইউ করে চলতে পারে কয়েকটি জরুরি ব্যাচবদ্ধ কাজ। আবার সরাসরি নিবিড় কাজের চাপ চলে আসা মাত্রই ব্যাচবদ্ধ কাজ স্থগিত থাকছে। এই নিষ্পলক দাসত্ব স্পার্টাকাসও দেখেনি।

প্রথম সত্যিকারের কেজো সময়ভাগ ব্যবস্থা চালু হয় ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনলজিতে, ১৯৬২-তে, সিটিএসএস (CTSS — **C**ompatible-**T**ime-**S**haring-**S**ystem)। বিশেষ রকমে বদলে নেওয়া আইবিএম ৭০৯৪ সিস্টেম ব্যবহার করা হয়েছিল এই সিটিএসএস-এর জন্যে। কিন্তু তখনো টাইমশেয়ারিং-এর চূড়ান্ত জনপ্রিয়তার প্রহর আসতে দেরি আছে, তার কারণ, প্রতিটি কাজকে তার নিজের রসদের মহলে যথোচিত মর্যাদার সঙ্গে প্রতিরক্ষা আর প্রাইভেসি দেওয়ার মত হার্ডওয়ার তখনো বাজারে ল্যান্ড করেনি। নতুন ধরনের হার্ডওয়ার বানানো শুরু হওয়ার সঙ্গে সঙ্গে, তৃতীয় জেনারেশনের কম্পিউটারেই, সময়ভাগ ক্রমে সমূহ জনপ্রিয় এবং পরিব্যাপ্ত হয়ে ওঠে । সিটিএসএস-এর সাফল্যের পর এবার এমআইটি, বেল ল্যাবরেটরি, জেনারেল ইলেকট্রিক সবাই লাইন করে লাগল একটা কম্পিউটার কুম্ভ নির্মাণের — যে একাই গোটা বুঁদির কেল্লা সামলে দেবে। তার শিরায় শিরায় সেঁটে দেওয়া হবে শ-শ হাজার হাজার ব্যবহারকারী, লোকপিছু এক পিস করে টেলিটার্মিনাল। ঠিক বিদ্যুৎব্যবস্থার মত, একটা কেন্দ্রীয় কম্পিউটার তার সীমাহীন অনন্ত ক্ষমতা নিয়ে খুশি করে চলবে গোটা জনগোষ্ঠীকে।

আমাদের আবহমান ঈশ্বরকেন্দ্রিক বা এসেন্সকেন্দ্রিক চিন্তাপ্রক্রিয়ার সঙ্গে চিন্তার এই গোটা মডেলটা এমন খাপে খাপে মিলে যায় যে আজো, সাইফির পর সাইফিতে, চোয়াল নামিয়ে গলার উচ্চারিত আদমের আপেলে ঠেকিয়ে, 'আয়াম ইওর ওয়ার্স্ট নাইটমেয়ার' বলতে বলতে, স্ট্যালোন তার ভিলেনব্যূহকে ফাইনাল চূড়ান্ত টার্মিনাল রকমের শায়েস্তা করে দেয় গোটা গ্রহ জুড়ে ছড়ানো কম্পিউটার ব্যবস্থার হৃদপিণ্ডস্বরূপ কেন্দ্রীয় কম্পিউটারকে চটকে দিয়ে। এখানে মার্কিন সমাজমননের কাম পুঁজি-ব্যবস্থার একমাবেদ্বিতীয়ম ঈশ্বর হয়ে ওঠার, সবাইকে প্রত্যেককে প্রশ্নহীন ভাবে নিয়ন্ত্রণ করার আদি ফ্যান্টাসিরও একটা ছায়া থাকে — তার সুপারকম্পিউটারের অগোচরে কোথাও একটা কেশও উৎপাটিত না-হওয়ার।

৮।। বহু নদীর মোহনা — মাল্টিক্স

বহু ব্যবহারকারীকে, মাল্টিপল ইউজারকে, একসঙ্গে কাজ করতে দেওয়ার এই নতুন কম্পিউটারের পরিকল্পিত ছকটা পরিচিত হল মাল্টিক্স বলে (MULTICS — **MULT**iplexed-**I**nformation-and-**C**omputing-**S**ervice)। গোটা বোস্টন শহরের প্রত্যেকটি ব্যবহারকারীর প্রত্যেকটি কাজ যে একাই সামলাবে। এই চিন্তার একটা বড় এলাকা জুড়ে ছিল সেই আমলের মেশিনের খরচকে প্রত্যেকটা ব্যবহারকারীর একটা সম্ভাব্য দূরত্বের মধ্যে নিয়ে আসা। একটা বড় মেশিন সবার ভাগে ভাগ হলে খরচ কমে যাচ্ছে, ইকনমি অফ স্কেল। কিন্তু শত চেষ্টাতেও এই বিজ্ঞানীদের কাছে সম্ভব ছিলনা ভবিষ্যতকে দেখা। এবং, একটা নাটকীয় ভাবে বদলে যাওয়া ভবিষ্যত। তাদের গোটা চিন্তাটার ভিত্তিতেই এই গলদটা রয়ে গেছিল। তখন কেউ যদি তাদের কাছে বলত যে সেদিনের সবচেয়ে বেশি শক্তির কম্পিউটারের চেয়েও বহু কোটি গুণ বেশি ক্ষমতার কম্পিউটার মাত্র তিরিশ বছরের মধ্যে সেদিনের দামের তুলনায় জলের দরে বিকোবে গোটা পৃথিবী জুড়ে — এর চেয়ে বড় সাইফি তাদের কাছে আর কিছু হতনা।

মাল্টিক্স-এর অভিজ্ঞতায় ভালো আর মন্দ দুই-ই ছিল। আজকে আমাদের কাছে অভাবনীয় লাগবে — ইন্টেল থ্রিএইটসিক্স মেশিনের চেয়ে সামান্য বেশি শক্তির একটা সিপিইউর সঙ্গে যুক্ত করা হয়েছিল কয়েকশো ব্যবহারকারীর টেলিটার্মিনালকে — যদিও তার ইনপুট/আউটপুট ক্ষমতা ছিল অনেকটা বেশি। নিজের সময়ের প্রকৌশলের সঙ্গে মাল্টিক্স-এর গরমিলটা অনেকটা চার্লস ব্যাবেজের অ্যানালিটিকাল ইঞ্জিনের মত। অবশ্য আজকের আমাদের কম্পিউটার কাজের সবচেয়ে আগ্রাসী রসদভুক মাল্টিমিডিয়া এবং জিইউআই তখন ছিলনা। জিইউআই বা গুই মানে গ্রাফিকাল ইউজার ইন্টারফেস। বলেছি আগেই, কিবোর্ডের বদলে ছবি আর ইঁদুর দিয়ে কাজ করা। কনসোলে কমান্ড-প্রম্পটে কমান্ড লিখে এন্টার মারার জায়গায় জাস্ট একবার ক্লিক করে ইঁদুরের পেট টিপছি। সঙ্গে সঙ্গে দেখছি চলৎচিত্র — ফাইলের ছবি উড়তে উড়তে একটা ফোল্ডারের ছবি থেকে আর একটা ফোল্ডারের ছবিতে গিয়ে

পড়ছে। এই চলৎ-চিত্র ভারি ভারি উঁচু দামে কেনা। গোপনে আমার মেশিনের যাবতীয় হার্ডওয়ার রসদের একটা সিংহভাগ নিরবচ্ছিন্নভাবে গিলে চলে এই গুই। এই গুই-এর ল্যাটাই ছিলনা মাল্টিক্স-এর আমলে। আর, আরো একটা বড় ব্যাপার এই যে, রসদের ওই টানাটানির ভিতর কাজ করতে গিয়ে তখনকার প্রোগ্রামচিরা অনেক দক্ষ ছোট সাইজের এবং অনেক কম রসদ-হ্যাংলা প্রোগ্রাম লিখতে জানতেন। যাই হোক, যত তৃতীয় বিশ্ব গোছের চাহিদা নিয়েই আসুক তারা, তাদের মোট চাহিদা মেটানো মাল্টিক্সকালীন হার্ডওয়ারের দ্বারা সম্ভব ছিলনা, এছাড়া 'পিএল' কম্পাইলার সংক্রান্ত কিছু সমস্যাও ছিল, তার কথায় আসছি — সব মিলিয়ে মাল্টিক্স লুপ্ত হল, কিন্তু তার পূজাবিধি লুপ্ত হলনা। মাল্টিক্স উত্তরাধিকার রয়েই গেল, নিজের সময়ের, রসদের, হার্ডওয়ারের তুলনায় বড্ড বেশি উচ্চাকাঙ্খী মাল্টিক্সের কাঙ্খিত হার্ডওয়ার যখন ক্রমে দেখা দিতে শুরু করল মর্তভূমিতে।

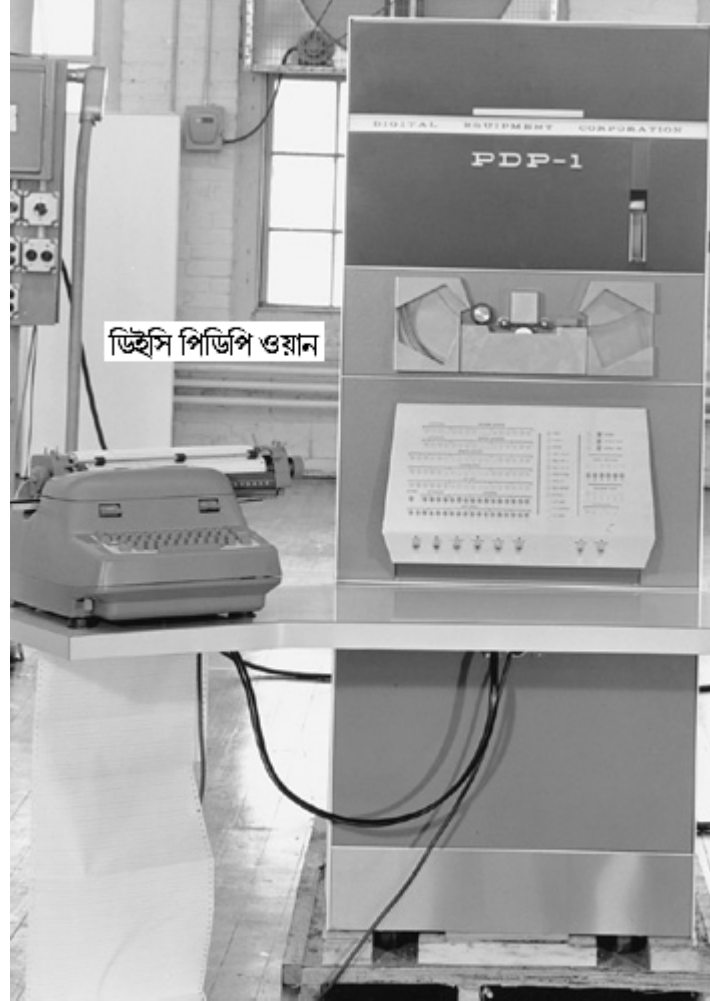
সামগ্রিক কম্পিউটার ভাবনায় কিছু দীর্ঘস্থায়ী গভীর মোচড় এনে দিল মাল্টিক্স, যাদের প্রায় গোটাটাই আমরা আজো বয়ে নিয়ে চলেছি। কিন্তু মাল্টিক্সকে ব্যবসায়িক সাফল্য করে তোলা গেলনা কিছুতেই। বেল ল্যাবরেটরি প্রোজেক্টটা ছেড়ে দিল, জেনেরাল ইলেকট্রিক তো কম্পিউটার-ব্যবসাই ছেড়ে দিল। কিন্তু এমআইটি প্রোজেক্টটা চালিয়ে গেল, এবং শেষ অব্দি মাল্টিক্স মেশিন খাড়াও করতে পারল। প্রায় আশিটা বড় কোম্পানি এবং বিশ্ববিদ্যালয়ে মাল্টিক্স ইনস্টল করা হয়েছিল। এই সব জায়গার ব্যবহারকারীরা মাল্টিক্স-এর প্রতি চূড়ান্তভাবে বিশ্বস্ত। জেনেরাল মোটরস, ফোর্ড, এবং নাসা তাদের মাল্টিক্স ব্যবহার বন্ধ করল এই সেদিন, নব্বই-এর দশকের শেষ দিকে — মাল্টিক্স ব্যবহার শুরু করার প্রায় তিরিশ বছর পরে।

ব্যবসায়িক সাফল্য না পেলেও মাল্টিক্স ধারণাটা এর পরবর্তী কম্পিউটার ডিজাইনগুলোকে এবং কম্পিউটার চিন্তাকে বদলে দিল। পরবর্তী কম্পিউটার প্রজন্মের উপর মাল্টিক্স-এর প্রভাব বিস্তারিত ভাবে নেটে আছে, নানা সাইটে, বিশেষ করে মাল্টিক্স-এর নিজের 'www.multicians.org' ওয়েবসাইটে। পরে আমরা এই নিয়ে আরো কথা বলব, গ্নু-লিনাক্স শিখতে চাইলে বলতেই হবে। ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সেই, যে কোনো ভৌত উপাদান বা ফিজিকাল ডিভাইসকেই দেখা হয় একটা ফাইল হিশেবে — ডিভাইসকে ফাইল হিশেবে দেখার এই ধারণাটা হল মাল্টিক্স-এর উত্তরাধিকার। হানিওয়েল বলে একটা প্রতিষ্ঠান, যারা জেনেরাল ইলেকট্রিকের কম্পিউটার ব্যবসার অংশটা কিনে নিয়েছিল, তারা এই মাল্টিক্স-এর ব্যবসায়িক সংস্করণও বাজারে ছাড়ে। তবে খুব একটা জনপ্রিয় হয়নি। মাল্টিক্স-এর আর একটা বড় অবদান কেন থম্পসন, যার কথায় আমরা এখুনি আসব, ইউনিক্স-এর প্রসঙ্গে। কেন থম্পসন তার প্রোগ্রামিং-এর কাজ শুরু করেছিলেন এই মাল্টিক্স দিয়েই। আগেই বলেছি, মাল্টিক্স-এর ব্যর্থতার আর একটা কারণ ছিল এর কুখ্যাত পিএল কম্পাইলার (PL/I — **P**rogramming-**L**anguage-I)। এই কম্পাইলার তৈরি করেছিল আইবিএম ১৯৬৪ থেকে ৬৯-এ। এতে ফোর্ট্রান কোবল, এবং অ্যালগল — তখনকার চালু এই তিনটে কম্পিউটার ভাষার মূল বৈশিষ্ট্যগুলোকে এক সঙ্গে নিয়ে আসা হয়েছিল, আর যোগ করা হয়েছিল ভুল-শোধরানোর ক্ষমতা এবং মাল্টিআস্কিং। কিন্তু এত বিতিকিচ্ছিরি রকমের জটিল হয়ে পড়ে এই পিএলআই, যে এতে কাজ করা প্রায় অসম্ভব ছিল। কেন থম্পসন এবং ডেনিস রিচির কৃতিত্বে এবার একটা মৌলিক বদল আসতে শুরু করে একদম এই প্রাথমিক জায়গা থেকেই। পরের সেকশনেই আসছি সেই কথায়।

৯।। মিনিকম্পিউটার, ইউনিক্স, সি

তৃতীয় জেনারেশন চলতে চলতেই মেশিনের বাজার নাটকীয় রকমে বদলে গেল। বাজারে বিপুল সংখ্যায় আসতে শুরু করল মিনি-কম্পিউটার। বড় সিস্টেমগুলোর থেকে এদের রিসোর্স বা রসদ বেশ কম, কিন্তু দাম বড় মেশিনের কুড়ি ভাগের এক ভাগ। বৈজ্ঞানিক আঙ্কিক কাজ যাদের নয়, তাদের জন্যে গন-উৎপাদিত হতে শুরু করল ছোট মেশিন। শুরু ১৯৬১-তে, ডিইসি পিডিপি এক থেকে, চলল এগারো অব্দি। এই পিডিপি ১-এর ক্ষমতা ছিল ১৮-বিটের শব্দের চার হাজার শব্দ। এর দাম ছিল এক লাখ কুড়ি হাজার ডলার, মানে আজকের, দোসরা ডিসেম্বর ২০০৩-এর চালু ডলার বিনিময় ৪৬.৩০ টাকার হিশেবে, পঞ্চান্ন লক্ষ ছাপ্পান্ন হাজার টাকা। কিন্তু, সেটাও সেই মুহূর্তে একটা ৭০৯৪-এর দামের পাঁচ শতাংশেরও কম। তাই, পটলডাঙার মুখে দ্বারিকের তেলেভাজার মত বিক্রি হল। সংখ্যা চিবোনোর কাজ বাদ দিলে অন্য কাজে পিডিপি সিরিজ গতিতে ৭০৯৪-এর প্রায় সমান ছিল।

পিডিপি সিরিজ নির্মাতা ডিইসি (Digital Equipment Corporation) তৈরি হয়েছিল ১৯৫৭-য়। যাটে এসে এরা কম্পিউটার তৈরি করা শুরু করে, কিন্তু মেশিনের নাম দেয় কম্পিউটার নয়, পিডিপি (PDP — **P**rogrammed-**D**ata-**P**rocessor)। নামটা দেখুন, অনেকটা মুড়িকে মুড়ি না বলে চাল-ভাজা বলে ডাকার মত। এর একটা পরিস্কার বেওসায়িক কারণ ছিল। আসলে মেশিনের ইমেজটা বদলাতে চাইছিল ডিইসি। কম্পিউটার বলতেই তখন লোকে বুঝত বিশেষ ধরনের বিল্ডিং, বিশেষ ধাঁচের কর্মচারী, বিরাট একটা স্কেল। ডিইসি এই গোটাটাকে সরিয়ে দিতে

ডিইসি পিডিপি ওয়ান

চাইছিল। আর ব্যবসা প্রতিষ্ঠানগুলোকে নিজের টার্গেট খদ্দের না-করে, তাক করেছিল বিজ্ঞান এবং ইলেকট্রনিক্স জগতের লোকেদের। এতে সফটওয়ার খরচও কমে গেছিল। এই লোকগুলো কম্পিউটারের কাজে অনেকটাই আত্মনির্ভর, নিজেদের পছন্দমত রকমে সফটওয়ার বানিয়ে নেবে, এবং বিক্রির পরে সহযোগিতাও অনেক কম করতে হবে। সাইবার বিজ্ঞানীরাও খুশি হয়েছিলেন চমৎকার খেলনা পেয়ে। নতুন নতুন প্রোগ্রাম লিখতে শুরু করেছিলেন। এই পিডিপি সিরিজের শেষটা ছিল পিডিপি ১১, যে মেশিনে বেল ল্যাবরেটরিতে কেন থম্পসন আর ডেনিস রিচির কাজ করার একটা ছবি দিয়েছি আমরা। যদিও, ওই ছবিটা আমি নিজেই জানিনা কপিরাইট আছে কিনা, বহু জায়গাতেই আছে, কিন্তু কম বাইটে ভালো রেজলিউশনের এটা পেয়েছি একটা জাপানি বা চিনা সাইটে। গুগলিতে ইমেজ সার্চ দিয়ে। কী ভাষা বুঝতে পারিনি, তাই ডিফল্টে ধরে নিতে অসুবিধে নেই যে, কপিরাইট নেই।

বেল ল্যাবরেটরিতে মাল্টিক্স সিস্টেম নিয়ে যে বৈজ্ঞানিকরা কাজ করছিলেন তাদের একজন ছিলেন কেন থম্পসন। ল্যাবরেটরিতে থম্পসন আপাত-দাবীদারহীন একটা মিনিকম্পিউটার হাতের কাছে পেয়ে গেলেন, যা সেই মুহূর্তে কেউ ব্যবহার করছে না, ওই পিডিপি সিরিজের পিডিপি ৭। একটা কাঁটছাঁট-করা, ছোটো সাইজের, একজন ইউজার বা ব্যবহারকারীর ব্যবহারের মাল্টিক্স অপারেটিং সিস্টেম কাঠামো বানিয়ে তোলার চেষ্টা শুরু করলেন থম্পসন এই মেশিনে। মাল্টিক্স লেখার ভাষা কিন্তু ওই পিএল-আই নয়, সরাসরি অ্যাসেম্বলি ল্যাংগুয়েজ। পিডিপি-সাত-এর কুচো

সাইজ আর সীমিত সামর্থ সত্ত্বেও, থম্পসন-এর এই চেষ্টা সত্যিই কাজ করেছিল। এবং এই কুচো সিস্টেমে দাঁড়িয়ে পরবর্তী বদল ও উন্নতির, ডিভেলপমেন্টের কাজটা করে যেতে পেরেছিলেন। থম্পসন কাজ করে চলাকালীন বেল ল্যাবরেটরির এক সহকর্মী ব্রায়ান কারনিঘান, মাল্টিক্স-এর নামের ধবনির সঙ্গে মিলিয়ে, মৃদু ক্যাওড়ামি সহ, এর নাম দেন ইউনিক্স (UNIX — UNICS — **Un**iplexed-**I**nformation-and-**C**omputing-**S**ervice)। মজাটা ছিল 'ইউনাক' বা খোজা শব্দটায়। মাল্টিক্স থেকে অনেককিছু গুরুত্বপূর্ণ ছেঁটে ফেলে একে পাওয়া গেছিল, ঠিক যেমন করে পাওয়া যেত প্রাসাদের হারেমের খোজাদের।

কেন থম্পসন এবং ডেনিস রিচি কাজ করছেন বেল ল্যাবের পিডিপি এগারোয়

পিডিপি ৭-এ এই গোড়াপত্তনের কিছুদিন পর কাজটাকে নিয়ে আসা হয় অনেকটাই উন্নততর একটা পিডিপি ১১-য়। বেল ল্যাবে কেন থম্পসনের সহকর্মীদের মধ্যে থেকে ডেনিস রিচি থম্পসনের এই প্রোজেক্টে যোগ দেন। এরপর যোগ দেন তার সমস্ত সহকর্মীই। এই শেষ মেশিনটা ছিল তার সময়ের তুলনায় খুবই এগোনো। মেমরির সাইজ ছিল দুই মেগাবাইট। এবং একটু আগে যে প্রতিরক্ষামূলক মেমরি প্রোটেকশন হার্ডওয়ারের কথা বলছিলাম আমরা, টাইমশেয়ারিং-এর প্রসঙ্গে, সেটাও ছিল এতে, যাতে একই সঙ্গে একাধিক ব্যবহারকারী একে ব্যবহার করতে পারেন। কিন্তু ১৬-বিট মেশিন হওয়ায় একটা একক প্রসেস বা পদ্ধতিতে আদেশ বা ইন্সট্রাকশনের জন্যে ৬৪ কিলোবাইট আর তথ্য বা ডেটার জন্যে ৬৪ কেবির বেশি থাকতে পারত না, ভৌত মেমরি অনেক বেশি হওয়া সত্ত্বেও।

এবার এল কী ভাষায় এই অপারেটিং সিস্টেমটা লেখা হবে সেই সমস্যা। এক নম্বর দিনে আমরা বলেছি, অপারেটিং সিস্টেমটাও একটা প্রোগ্রাম, গোটা ব্যবস্থার সবচেয়ে রেলাবাজ প্রোগ্রাম। তার চলার উপর দাঁড়িয়ে চলতে পারে অন্য আর সব যাবতীয় প্রোগ্রাম। এবং সময়টা ভাবুন, তখন অপারেটিং সিস্টেমও বানিয়ে নিতে হয়, আজকের মত হাতে গরম তৈরি কারনেল পাওয়া যায়না, যা মোটামুটি যে কোনো হার্ডওয়ারেই চলে। তার মানে, একটা মেশিন ছেড়ে যদি আপনার কাজ আপনি অন্য মেশিনে নিয়ে যেতে চান, তাহলে সেই কাজের পিছনের অ্যাপ্লিকেশন প্রোগ্রামটা আপনাকে সেই নতুন মেশিনে চালাতে হবে। সেটা চালাতে গেলে তার আগে চালাতে হবে সকল প্রোগ্রামের নবাব প্রোগ্রাম কারনেল বা অপারেটিং সিস্টেমকে। কিন্তু পুরোনো মেশিনের অপারেটিং সিস্টেম তো নতুন মেশিনে চলবে না, তাদের হার্ডওয়ার আলাদা হতে পারে, আইও ডিভাইস আলাদা হতে পারে, সেগুলোর আলাদা ড্রাইভারকে মিলিয়ে নতুন অপারেটিং সিস্টেম গজিয়ে নিতে হবে নতুন মেশিনে, যাতে এবার তার উপর দাঁড়িয়ে আপনার কাজের অ্যাপ্লিকেশন প্রোগ্রামটা আপনি চালাতে পারেন। এটাই স্থানান্তর বা ট্রান্সপোর্টেবিলিটির সমস্যা, চালু টেকনিকাল শব্দে 'পোর্ট করা' বলে ডাকা হয়, পরে একাধিকবার আমরা আসব এই প্রসঙ্গে।

তদ্দিনে মেশিন থেকে মেশিনে প্রত্যেকবার নতুন করে অপারেটিং সিস্টেম লিখে লিখে প্রোগ্রাম ডিভলপাররা হাড়ে হাড়ে বুঝেছেন এমন একটা অপারেটিং সিস্টেমের মাহাত্ম্য যা সব মেশিনেই সমান পটীয়ান হবে। থম্পসন নিজেই বানিয়েছিলেন একটা কম্পিউটার ভাষা, তার নাম বি। এই বি নামটা এসেছিল এর আগের একটা ভাষা বিসিপিএল-এর উত্তরাধিকার থেকে। বিসিপিএল (BCPL — **B**asic-**C**ombined-**P**rogramming-**L**anguage) এসেছিল

সিপিএল থেকে, ঠিক ওই পিএল-আই-এর মতই আর একটা ভাষা যা কোনোদিনই তার প্রতিশ্রুতিমত কাজ করে উঠতে পারেনি। প্রথমে থম্পসন ইউনিক্স-কে এই বি-তে লেখার কথা ভেবেছিলেন। কিন্তু বি-র নিজস্ব কিছু সমস্যার জন্যে এই চেষ্টা সফল হলনা। তখন ডেনিস রিচি এই বি-র একটা বংশধর বানালেন, তার নাম আর কী হতে পারে — বি-এর পর, সি ছাড়া? আর লিখলেন এই সি-এর একটা ভালো কম্পাইলার। এরপর, কেন থম্পসন আর ডেনিস রিচি মিলে ইউনিক্স নামের অপারেটিং সিস্টেমটাকে সি-তে লিখে ফেললেন।

বিভিন্ন বিশ্ববিদ্যালয় থেকে বিজ্ঞানীরা এবার ইউনিক্স চাইতে লাগল বেল ল্যাবরেটরি তথা রিচি-থম্পসনের কাছে। মার্কিন সরকারের সেই সময়কার একটা আইন মোতাবেক, বেল ল্যাবরেটরির মূল কোম্পানি এটিঅ্যান্ডটি-র পক্ষে কম্পিউটার সংক্রান্ত কোনো ব্যবসায় নামা সম্ভব ছিলনা। তাই, নামমাত্র একটা টাকায় এই ইউনিক্স পৌঁছে গেল বিশ্ববিদ্যালয়গুলোয়। আর পিডিপি সিরিজের বিজ্ঞানীচাটা ব্যাপারটা তো বলেছি আগেই। বিজ্ঞানীরা তাদের মনের মত একটা সিস্টেম পেয়ে গেলেন, কম্পিউটারের ইতিহাসে সবচেয়ে সুদূরপ্রসারী বদলগুলোর একটা জোড়া বদল ঘটে গেল একই সাথে — ইউনিক্স আর সি। ইতিহাস আসলে এই ভাবেই গজায়, কতকগুলো পরপর ঘটতে থাকা সমাপতন মিলে ফুটে ওঠে একটা ধাঁচ, প্রায় অলক্ষে, প্রায় অকারণে। মানে গূঢ়তর কোনো হেতু ব্যতিরেকেই। পরে আমরা, জ্যান্ত ঘটনা মরে ইতিহাস হয়ে গেলে, পড়ি যখন, চমকাই, আরে তাইতো — এই ইউনিক্স-এর বেলাতেও এইরকম একটা সমাপতন খুব গুরুত্বপূর্ণ জায়গা নিয়েছিল। এই বিশ্ববিদ্যালয়গুলোর অ্যাকাডেমিক আবহাওয়ায় বেশিরভাগ কেসেই মেশিনটা ছিল পিডিপি ১১, যাতে বানানো ইউনিক্স। তাই অনেক বেশি অনাবিলতায়, প্রায় কোঁৎ না-পেড়েই ইউনিক্স চালানো যেত এই মেশিনগুলোয়। ইউনিক্স-এর পতাকা পৎপৎ নয়, দুমদাম করে উড়তে শুরু করল। আর সি-তে লেখা মানে এই ইউনিক্স এবার একটা মেশিন থেকে অন্য মেশিনে নিয়ে যাওয়া, যার অন্য নাম পোর্ট করা, ভারি সহজ হয়ে গেল। শুধু ওই সি কম্পাইলারটুকুই লিখতে হবে নতুন মেশিনের মত করে, তারপরেই তাতে সি চালানো যাবে। আর সি চালাতে পারা মানেই তাতে ইউনিক্স চালাতে পারা।

ইউনিক্সের পরবর্তী প্রজন্মের বিভিন্ন বাণিজ্যিক ক্লোন যারা পোজিক্স মেনে তৈরি

নাম	নির্মাতা	মন্তব্য
জেনিক্স (Xenix)	মাইক্রোসফট	এখন অবলুপ্ত
বিএসডিআই (BSDi), বিএসডি অপারেটিং সিস্টেম (BSD/OS), ফ্রি-বিএসডি (FreeBSD)	বার্কলে সফটওয়ার ডিজাইন	বার্কলে সফটওয়ার ডিস্ট্রিবিউশন যারা করেছিলেন তাদেরই বাণিজ্যিক প্রয়াস
সান অপারেটিং সিস্টেম (SunOS)	সান মাইক্রোসিস্টেম	পরে সোলারিস-এর মধ্যেই ঢুকে যায়
সোলারিস (Solaris)	সান মাইক্রোসিস্টেম	পিসিতে চালানোর সংস্করণও আছে
এইচপি-ইউএক্স (HP-UX)	হিউলেট প্যাকার্ড	
আইক্স (AIX)	আইবিএম	
আলট্রিক্স (Ultrix)	ডিইসি	অবলুপ্ত, পরে ডিজিটাল ইউনিক্স এর জায়গা নেয়
ডিজিটাল ইউনিক্স (Digital UNIX)	ডিইসি	
আইরিক্স (IRIX)	সিলিকন গ্রফিক্স	
স্কো ওপন সার্ভার (SCO Open Server)	সান্টা ক্রুজ অপারেশন	পরে স্কো-ইউনিক্সওয়ার এর জায়গা নেয়
স্কো ইউনিক্সওয়ার (SCO UnixWare)	সান্টা ক্রুজ অপারেশন	পিসিতে চালানো যায়

তথ্যসূত্র — সুমিতাভ দাসের 'ইউনিক্স দি আলটিমেট গাইড'

১০।। ইউনিক্স, মিনিক্স, লিনাক্স

আমরা এক নম্বর দিনে বলেছি, উইন্ডোজ অপারেটিং সিস্টেমের ক্ষেত্রে সোর্স-কোড উন্মুক্ত নয়। এর একটা আন্দাজ আমরা ইতিমধ্যেই পেয়েছি, তাও একটু ঝালিয়ে নিই এবার। কম্পিউটারে যাদের নিয়ে আমরা কাজ করি সেই অ্যাপ্লিকেশনগুলো প্রোগ্রাম, আবার তাদের কাজের ভিত্তি মানে অপারেটিং সিস্টেমও একটা প্রোগ্রাম। এই প্রোগ্রামগুলো সরাসরি বানানো যায়না। প্রোগ্রামটাকে বানাতে গিয়ে, প্রথমে লেখা হয় মানুষের বোধ্য রকমে, হাই

লেভেল কোনো কমপিউটার ভাষায়, যখন তার নাম কোড। ইউনিক্সের ক্ষেত্রে কোডের ভাষাটা প্রায় গোটাটাই সি। এই সি-তে বা অন্য ভাষায় লেখা কোডটা তারপর কম্পাইল করা হয়, মানে, কম্পিউটারের বোধ্য আদেশমালায় রূপান্তরিত করা হয় — লো লেভেল ভাষায়। লেভেলটা সত্যিই ভারি লো, আগেই যেমন বলেছি, কম্পিউটার তো সত্যিই বেশ কম বোঝে, সাধে আমরা গাধা বলে ডেকেছিলাম। তখন সেই কম্পাইলড কোডটা হয়ে দাঁড়ায় একটা প্রোগ্রাম। এই কম্পিউটার-বোধ্য আদেশমালায় বা প্রোগ্রামে রূপান্তরিত বা কম্পাইলড হওয়ার পর মূল মানুষবোধ্য কোডটা এবার একটা নতুন আকার পায়, এমন ফাইল যা নিজে নিজে চলে। ইউনিক্স বা সোর্স-উন্মুক্ত গ্নু-লিনাক্স জগতে এই ফাইলগুলোকে ডাকা হয় বাইনারি ফাইল বলে, আর উইন্ডোজ জগতে এক্সিকিউটেবল ফাইল। সেটা ইউনিক্স জগতের বাইনারি হোক, বা উইন্ডোজ জগতের এক্সিকিউটেবল, তাদের প্রত্যেকেরই তাহলে অন্য একটা আকার থাকে — একটা সোর্স কোড, মানুষবোধ্য মূল আকার, যাকে কম্পাইল করে এই চালনীয় আকারটা আসে।

আমরা যখন মাইক্রোসফট উইন্ডোজ কিনি, বা মাইক্রোসফটের কোনো প্রায়োগিক প্রোগ্রাম, এমএস-ওয়ার্ড জাতীয়, মাইক্রোসফট আমাদের ওই কম্পাইলড চালনীয় আকারটা বিক্রি করে, কখনো তার মূল আকার বা সোর্স কোডটা জানতে দেয়না। সিডি-তে লেখা সেই কম্পাইলড আকারটা আমরা নিজেদের মেশিনে চালাই। মূল প্রোগ্রামটা আমাদের জানার বা দেখার সুযোগ থাকেনা। গ্নু-লিনাক্স জগতকে ওপন-সোর্স জগত বলা হয় কারণ সেখানে কম্পাইলড আকারটা যেমন পাওয়া যায়, মূল প্রোগ্রামটাও পাওয়া যায়, যাকে আমরা কম্পাইল করতে পারি নিজের মেশিনে, বা বদলাতে পারি, যদি দরকার পড়ে। অর্থাৎ যে একটা নতুন অপারেটিং সিস্টেম বা অ্যাপলিকেশন সফটওয়ার বানাতে চায় সে ওই উন্মুক্ত সোর্স কোড থেকে কাজ শুরু করতে পারে।

কিন্তু সোর্স কোড যদি গোপন বা ক্লোজড রাখা হয় সেক্ষেত্রে যে নতুন কিছু বানাতে চায় তাকে শুরু করতে হয় শূন্য থেকে, মানে মাইক্রোসফট যেখান থেকে শুরু করেছিল, তারপর মাইক্রোসফটের আবিষ্কৃত প্রতিটি পদক্ষেপ তাকে আবার নতুন করে আবিষ্কার করে নিতে হয়। আর এক ভাবে ভাবুন, মাইক্রোসফট কী? একটা পুঁজির নাম, যে কয়েকজন প্রতিভাবান প্রোগ্রামারকে পয়সার বিনিময়ে নিয়োগ করছে, তাদের কাজটাকে বাজারে নিয়ে আসছে নিজের ব্রান্ডনেম লাগিয়ে। ওই প্রোগ্রামারদের পরের প্রজন্ম তাদের পূর্বপুরুষের কাজের থেকে বঞ্চিত হল, তাদের ইতিমধ্যেই হেঁটে ফেলা পথ আবার করে হাঁটতে হবে। এটাকেই রিচার্ড স্টলম্যান বলেছেন রি-ইনভেন্টিং দি হুইলস, বা বারবার করে চাকা আবিষ্কার করে চলার চক্কর।

রিচার্ড স্টলম্যান

এই প্রসঙ্গটা এখানে আনতে হল, কারণ, প্রথম যখন থম্পসন এবং তার সঙ্গীরা মিলে ইউনিক্স বানান, সেটা ছিল ওপন সোর্স। যার কাজে লাগবে সেই সেটা পেয়েছে, পরে যখন বিভিন্ন ব্রান্ডনেম-এর ইউনিক্স বাজারে এল সেগুলো আর ওপন-সোর্স নয় (আরো বিশদ করে এটা জানব আমরা পাঁচ নম্বর দিন)। কেন থম্পসন-রা যেহেতু সোর্স কোডটা যে কারুর জন্যে উন্মুক্ত করে দিয়েছিলেন, তার থেকে বিভিন্ন প্রতিষ্ঠান বা বিশ্ববিদ্যালয় বা গবেষণা-সংস্থা তাদের নিজেদের মত ইউনিক্স সংস্করণ তৈরি করে নেয়, তাদের প্রয়োজনমাফিক সেই মূল কোডকে বদলিয়ে। কিন্তু এতে বিভিন্ন বিচিত্র রকমের ইউনিক্সের একটা খিচুড়ি তৈরি হয়ে যায় — তাদের একটা নিয়মনিষ্ঠ আকার দিতে গিয়ে পরে দুটো মূল ভার্শন আসে। একটা সিস্টেম-ভি, আর অন্যটা বিএসডি (বার্কলে সফটওয়ার ডিস্ট্রিবিউশন)।

এই ভার্শনগুলোর মধ্যে কিছু কিছু ছোটখাট তফাতও ছিল। একটা ইউনিক্সে তৈরি করা প্রোগ্রাম যাতে অন্য যে কোনো ইউনিক্সে চালানো যায় তার জন্যে তৈরি হল পোজিক্স (POSIX — **P**ortable-**O**perating-**S**ystem-**I**nterface) — প্রায়োগিক প্রোগ্রামগুলোর সঙ্গে কম্পিউটার হার্ডওয়ারের এবং অপারেটিং সিস্টেমের পারস্পরিক সম্পর্কের কিছু নির্দিষ্ট মাপকাঠি, কিছু মানদণ্ড তৈরি করে দেওয়া, যাতে প্রতিটি অপারেটিং সিস্টেমে আলাদা আলাদা করে সোর্স কোড তৈরি করতে না-হয়, একবার তৈরি করেই তা নানা অপারেটিং সিস্টেমে কম্পাইল করে নেওয়া যায়। পোজিক্স ছিল ইউনিক্স ঘরানার। পরে গ্নু-লিনাক্স জগতের এক্স/ওপন (X/Open Specifications) ব্যবস্থা পোজিক্স-এর উপরে নির্ভর করে তৈরি হয়েছিল। দুই নম্বর দিনে আমরা সিস্টেম-কলের কথা বলেছি, একটা সফটওয়ার যখন কোনো একটা রসদ ব্যবহার করতে দেওয়ার আবেদন জানায় অপারেটিং সিস্টেমের কাছে, সেই সিস্টেম-কলের কিছু নির্দিষ্ট ছক ঠিক করে দেয় পোজিক্স। নিজের ব্যবস্থাকে ইউনিক্স বলে ঘোষণা করতে গেলে যে ছক মানতে হবে। যাতে অন্যদের কাজের সঙ্গে আমার সিস্টেমের একটা সাযুয্য বা কম্পেটিবিলিটি আসে।

অ্যান্ড্রু ট্যানেনবম

১৯৮৭-তে ট্যানেনবম তার ছাত্রদের কম্পিউটার তত্ত্ব বোঝানোর প্রয়োজনে তৈরি করেন মিনিক্স বলে একটা অপারেটিং সিস্টেম। এটা এক অর্থে ছিল ইউনিক্স-এর একটা প্রতিরূপ — ক্লোন। ইউনিক্সের সঙ্গে মিনিক্সের ব্যবস্থা প্রায় হুবহু এক, শুধু আকারে অনেক ছোট। এবং মিনিক্স-ও এই পোজিক্স ব্যবস্থা মানে। গোটা সোর্স কোড সহ এই মিনিক্স অপারেটিং সিস্টেম এবং তার প্রয়োজনীয় পাঠ্য বিনামূল্যে নেটে পাওয়া যায়, 'www.cs.vu.nl' সাইটে। কিন্তু মিনিক্স-এর মূল ঝোঁক ছিল লেখাপড়া, বিদ্যাচর্চা। তাকে কেজো জায়গায় এনে অন্য অপারেটিং সিস্টেমের বিকল্প হিশেবে ব্যবহার করা প্রায় অসম্ভব। মিনিক্স-এর মডেলে, মিনিক্স ব্যবহার করে এই কাজটা শুরু করেন ফিনল্যান্ডের এক ছাত্র, লিনাস টরভাল্ডস। রিচার্ড স্টলম্যান তার বেশ কিছু বছর আগে থেকেই শুরু করে দিয়েছিলেন তার মুক্ত সফটওয়ার আন্দোলন, এবং তার সংগঠন আর ওয়েবসাইট, গ্নু (GNU — **G**NU's-**N**ot-**U**NIX)। এই আন্দোলনের অংশ হিশেবেই এসেছিল স্টলম্যানদের বানানো কম্পাইলার জিসিসি। আর মেক, অটোকনফ ইত্যাদি সফটওয়ার। যাকে ব্যবহার করেন লিনাস তার অপারেটিং সিস্টেমের কম্পাইলেশনে। এবং স্টলম্যানদের এই কাজগুলো সবই জিপিএল বা জিএনইউ পাবলিক লাইসেন্সের আওতায়। যা প্রচলিত কপিরাইট আইনের বিরোধী, কিন্তু তার মানে কপিরাইট ছেড়ে দেওয়া নয়, অন্য ধরনের একটা কপিরাইট।

লিনাস টরভাল্ডস

ধরুন আপনি একটা সফটওয়ার বানালেন। সেটা আপনি জিপিএলের আওতায় আনলেন। তার মানে এই নয় যে আপনি সেটা বিনামূল্যে দেবেন। আপনি সেটা বিক্রি করতেই পারেন। কিন্তু সেই সফটওয়ার যদি কেউ বিনামূল্যে দেয়, তাহলে আপনার কোনো বাধা দেওয়ার অধিকার থাকবে না। বা, সেই মূল সফটওয়ার ব্যবহার করার এবং বদলানোর অধিকার থাকবে যে কারুরই। কিন্তু যেহেতু সেটা একটা জিপিএল-এর আওতায় থাকা সফটওয়ার ব্যবহার করে বানানো, জিপিএল মোতাবেক এই নতুন বদলে তোলা সফটওয়ারটাও আপনা থেকেই জিপিএল-এর আওতায় চলে আসবে। এবার, যে এটা বদলালো সে এটা বিক্রি করতেই পারে। কিন্তু বদলানো সংস্করণ অন্য যে কেউ যদি বিনামূল্যে দেয় তাকে আটকানোর অধিকার তার থাকবে না।

অর্থাৎ, জিপিএল-এর মধ্যে বাজারের চালু কাঠামোর বিরুদ্ধে লড়াই-এর একটা বুদ্ধিদীপ্ত বজ্জাতি আছে — যে কারণে তিনি গোটা ওপন সোর্স জগতের কাছে ধন্যবাদার্হ। একবার জিপিএল-এর আওতায় কোনো সফটওয়ার আসা মানেই সেখান থেকে এটা শেষহীন শৃঙ্খল চালু হয়ে যাওয়া। তাই, একবার জিএনইউ-র কোনো সফটওয়ার ব্যবহার করা মানে গোটা লিনাক্স সফটওয়ারের জগতটাই এসে গেল জিপিএল-এর আওতায়। লিনাস টরভালডস-এর দেওয়া ওই অপারেটিং সিস্টেম সহ। তৈরি হল গ্নু-লিনাক্স। কিন্তু গ্নু-লিনাক্স-এর এই পুরো ঘটনাটাই অনেক পরের। চতুর্থ জেনারেশন কম্পিউটারের সময়ে এসে। পাঁচ নম্বর দিনে এবং পরেও আবার এই প্রসঙ্গে আসব আমরা।

১১।। চতুর্থ জেনারেশন কম্পিউটার — ১৯৮০ থেকে চলছে ...

তৃতীয় জেনারেশন থেকে এবার চতুর্থ জেনারেশন কম্পিউটার আসায় একটা বড় ভূমিকা নিল এলএসআই বা লার্জস্কেল ইন্টিগ্রেটেড সার্কিট। যেখানে এক একটা চিপের এক বর্গ সেন্টিমিটার সিলিকনে ভরা থাকে হাজার হাজার ট্রানজিস্টর। এই এলএসআই-এর হাত ধরেই পৃথিবীতে এল পার্সোনাল কম্পিউটার বা পিসির জমানা। গঠন কাঠামোর দিক থেকে এই পিসি, প্রথমে যাকে ডাকা হত মাইক্রোকম্পিউটার বলে, এর আগের তৃতীয় জেনারেশনের মিনি-কম্পিউটারদের থেকে কিন্তু আদৌ তেমন আলাদা ছিলনা। প্রাথমিক নাটকীয় পার্থক্যটা এল দামের নিরিখে। একটা মিনি কিনতে একটা কোম্পানির একটা বিভাগ নিদেন বিশ্ববিদ্যালয়ের একটা ডিপার্টমেন্ট লাগতই, সেখানে এই মাইক্রোপ্রসেসর চিপের কল্যাণে একজন ব্যক্তিও এবার একটা কম্পিউটার কেনার কথা ভাবতে শুরু করতে পারল।

১৯৭৪-এ, ইনটেল এল তার ৮০৮০ নিয়ে, তার সিপিইউ কাজ করত ৮ বিটে, এবং মোটামুটি ভাবে একটা বিস্তৃত পরিসরের যে কোনো কাজেই ব্যবহার করা যেত ৮০৮০-কে। এই ৮০৮০-র জন্যে ইনটেলের এবার দরকার পড়ল একটা অপারেটিং সিস্টেম বা ওএস। এই ওএস লাগাতে গিয়ে মেশিনে যুক্ত করা হল তখন সদ্য বাজারে আসা ৮ ইঞ্চি ফ্লপি ডিস্ক, তার জন্যে বানানো হল সিপি/এম (CP/M — **C**ontrol-**P**rogram-for-**M**icroarchitecture) নামে একটা ডিস্ক অপারেটিং সিস্টেম। আমরা ডস (DOS — **D**isk-**O**perating-**S**ystem) বলে যাকে চিনি তার তখনো ধরাধামে অবতীর্ণ হওয়ার দেরি আছে। ইনটেল এই গোটা কাজটার দায়িত্ব দিয়েছিল তাদের অন্যতম কনসালট্যান্ট গ্যারি কিল্ডলের উপর। কিল্ডল এবার এই সিপি/এম অপারেটিং সিস্টেমের বিক্রি এবং বিকাশের জন্য নিজের একটা প্রতিষ্ঠান খোলেন, ডিজিটাল রিসার্চ। ১৯৭৭-এর ভিতরেই ডিজিটাল রিসার্চ সিপি/এম-কে সেই সময়ের প্রায় সবগুলো মাইক্রোকম্পিউটারে ব্যবহারযোগ্য করে তোলে। এবার, এই সিপি/এম-এর উপর ভিত্তি করে অনেক অ্যাপ্লিকেশন প্রোগ্রাম বানানো শুরু হয়। শুধু ডিজিটাল রিসার্চ নয়, এতে সামিল হয় আরো অনেক কোম্পানি।

৮০-র দশকের একদম গোড়ায় আইবিএম বাজারে আসে আইবিএম-পিসি নিয়ে। আইবিএম তখন এই পিসির উপযোগী ওএস এবং অ্যাপ্লিকেশন সফটওয়ার খুঁজছে। এর আগেই মাইক্রোসফট-কুখ্যাত বিল গেটস, তার বেসিক (BASIC) নামের ইন্টারপ্রিটার লিখে ফেলেছেন, জন কেমেনি আর থমাস কুর্টজ-এর প্রাথমিক সংস্করণটাকে ব্যবহার করে। মূলত ষাটের দশকে ডার্টমুথ-এর টাইমশেয়ারিং কম্পিউটারগুলোর জন্যে লেখা কম্পিউটার ভাষা বেসিক কাজ করে কম্পাইলার নয়, ইন্টারপ্রিটার দিয়ে। পার্থক্যটা আমরা পরে বুঝব। কম্পিউটার ভাষাকে অকারণ জনপ্রিয় এবং সহজ করতে গিয়ে বেসিক-এ যা করা হয়েছে তা হল, কোনো কোনো প্রোগ্রামিং-শিক্ষকের মতে, পরবর্তীকালে কোনো সিরিয়াস কম্পিউটার ভাষায় কাজ করার ক্ষমতা শেষ করে দেওয়া।

নিজেদের পিসি বাজারে আনার আগে আইবিএম বিল গেটস-এর সঙ্গে যোগাযোগ করে, যদি পিসির উপযোগী কোনো ওএস তার জানা থাকে। যোগাযোগ করা হয় ডিজিটাল রিসার্চের সঙ্গেও। কিন্তু আইবিএম পিসি কিভাবে

পরবর্তী যুগে কম্পিউটার চর্চার নিয়ামক হয়ে উঠবে সেটা অন্য অনেকেরই মত কিল্ডলও আন্দাজ করতে পারেননি। তিনি আইবিএম-কে কোনো পাত্তাই দিলেন না। বিল গেটস দিলেন। অনেকটা সিপি/এম-এর মতই আর একটা ডিস্ক অপারেটিং সিস্টেম বানিয়েছিল সিঅ্যাটল কম্পিউটার প্রোডাক্ট নামে একটা কোম্পানি। যার নাম ডস, ডিস্ক অপারেটিং সিস্টেম। এদের কাছ থেকে গেটস কিনে নিলেন ডস, পরবর্তীকালে মাইক্রোসফটের ব্যবসার আয়তনের তুলনায় আক্ষরিকভাবেই মাত্র খুঁদকুড়োর দামে। মাত্র ৫০,০০০ ডলারে। ওই বেসিক আর এই ডস-কে একসাথে পিসির সফটওয়ার প্যাকেজ হিশেবে আইবিএম-এর কাছে বিক্রি করলেন গেটস। আইবিএম এই মূল প্যাকেজে কিছু বদল ঘটাতে চাইল, তার জন্য গেটসকে লোক ভাড়া করে আনতে হল সেই সিঅ্যাটল কম্পিউটার প্রোডাক্ট থেকেই। এই বদলানো সংস্করণের নাম দেওয়া হল এমএস-ডস। মাইক্রোসফট-ডস। অচিরেই আইবিএম তথা অন্য পিসির দুনিয়া জয় করে নিল এমএস-ডস। মূলত তার প্রায়োগিক সহজতার জন্যে। এখানে কাজ করল গেটসের দ্বিতীয় ব্যবসায়-বুদ্ধি। তিনি এমএস-ডসকে বিক্রি করতে শুরু করলেন বিভিন্ন কম্পিউটার নির্মাতার কাছে, যার কাছে হেরে গেল কিল্ডলের রণনীতি, নির্মাতাকে না-করে, সরাসরি ব্যবহারকারীকে বিক্রি করতে চেয়েছিলেন।

৮১-র পিসি/এক্সটি-র জায়গায় ১৯৮৩-তে আইবিএম বাজারে আনল পিসি/এটি, যাতে মাইক্রোপ্রসেসর ব্যবহৃত হল ইনটেল ৮০২৮৬, এবং ব্যবহৃত হল এটি-বাস, যাতে তথ্য চলাচল করবে ১৬ বিটের এককে, মানে পিসি/এক্সটি-র চেয়ে উন্নততর (AT — **A**dvanced-**T**echnology)। এমএস-ডস ইতিমধ্যেই সুপ্রতিষ্ঠিত এবং সিপি/এম প্রায় অবলুপ্ত। এর পরে ইনটেল যে ৮০৩৮৬ এবং ৮০৪৮৬ মেশিন আনল, তাতেও ব্যাপকভাবে ব্যবহৃত হল এমএস-ডস। এমএস-ডসের প্রথম দিককার সংস্করণগুলো ক্রমে অনেক সংস্কৃতও হল। অনেক উন্নত বৈশিষ্ট্য তুলে আনা হল ইউনিক্স থেকে।

১২।। গ্রাফিকাল ইউজার ইন্টারফেস এবং নেটওয়ার্কিং

এই সিপি/এম বা এমএস-ডস, ঠিক গ্নু-লিনাক্স এর ব্যাশ শেলের মতই, কমান্ড প্রম্পটে কাজ করতে থাকা একটা কমান্ড ইন্টারপ্রিটার। একটা কমান্ড লাইন থাকে, যেখানে একটা কার্সর দপদপ করতে থাকে। সেখানে কিবোর্ড থেকে কমান্ড টাইপ করে তোলা হয়, তারপর এন্টার চাবিটা টেপা মাত্র কমান্ডটা পৌঁছে যায় কম্পিউটারের কাছে — কম্পিউটার কাজ শুরু করে। এই কমান্ড-প্রম্পট থেকে ইঁদুর আর ছবির রাজত্ব মানে গ্রাফিকাল ইউজার ইন্টারফেস বা জিইউআই বা গুই এসেছিল প্রাথমিক ভাবে একটি মানুষের গবেষণার উপর নির্ভর করে। স্ট্যানফোর্ড রিসার্চ ইনস্টিটিউটের ডাগ এঙ্গেলবার্ট ষাটের দশকের মাঝামাঝি তৈরি করেন এই গুই। এর মধ্যে পড়ে উইনডোজ বা কাজ করার জন্যে একটা জানলার ছবি, তার সঙ্গে আরো নানা উপাদান, মেনু, আইকন, বোতাম ইত্যাদি — মাউস ব্যবহার করার নানা তরকিব। এঙ্গেলবার্ট-এর এই কাজকে ঘিরে তারপর ব্যাপক গবেষণা এবং বিকাশের কাজ শুরু হয়। অ্যাপল কোম্পানির স্টিভ জবস এই গুই নিয়ে আসেন অ্যাপল-এ এবং তৈরি হয় অ্যাপল ম্যাকিনটশ। গুই পিসি-র ইতিহাসে আর একটা অধ্যায় নিয়ে আসে, যার অন্য নাম 'ইউজার ফ্রেন্ডলি'। মানে, ব্যবহারকারী কম্পিউটার বিজ্ঞান কিছু না-জেনেই, ব্যবহারের কিছু রীতি শিখে কম্পিউটার ব্যবহার করতে পারবে, সফটওয়ার প্রয়োগ করতে পারবে তার বিভিন্ন কাজে। এই ব্যবহারকারী শুধু যে কম্পিউটার বিজ্ঞান কিছু জানে-না তাই নয়, জানার কোনো বাসনা বা পরিকল্পনাও তাদের নেই। অর্থাৎ প্রথম যুগের মেশিন যারা ব্যবহার করত সেই প্রোগ্রামার-অপারেটর ইউজার থেকে সম্পূর্ণ নতুন ধরনের একটা ব্যবহারকারী তৈরি হল এবং বাড়তে লাগল — কনজিউমার-ইউজার বা ভোক্তা-ব্যবহারকারী। একটা ছোট সংখ্যক প্রোগ্রামার-ইউজারের দায়িত্ব রইল সফটওয়ার বানানো বদলানো বোঝা এবং চর্চা। আর বৃহত্তর গোষ্ঠী তৈরি হল কনজিউমার-ইউজারের, যারা টিভি-সিডি-প্লেয়ার-ডিভিডি-প্লেয়ার-গেমস-কনসোল-নেট-ব্রাউজার ইত্যাদি বিভিন্ন রকমে ভোগ করবে কম্পিউটারকে, কম্পিউটার বিজ্ঞানের কোনো তোয়াক্কা না-রেখেই।

মাইক্রোসফট যখন এমএস-ডসের পরবর্তী প্রজন্ম নিয়ে কাজ শুরু করল, অ্যাপল ম্যাকিনটোশের জনপ্রিয়তা তাদের ভীষণভাবে প্রভাবিত করল। তৈরি হল একটা গুই-ভিত্তিক ওএস, এমএস-উইনডোজ। প্রথম দিককার উইনডোজ চালাতে হত ডস থেকেই, কমান্ড প্রম্পট-এ 'উইন' (win) টাইপ করে এন্টার মারতে হত। উইন্ডোজ তখনো ছিল একধরনের একটা গুই খোলস, ডস ওএস-এর উপরে চলত। প্রায় দশ বছর ধরে, ১৯৮৫ থেকে ১৯৯৫ এই অবস্থা চলেছিল। ১৯৯৫-এ এলো স্বনির্ভর একটা গুই সিস্টেম, উইনডোজ নাইন্টিফাইভ। এখানে আভ্যন্তরীন এমএস-ডস

ব্যবস্থাটা ব্যবহৃত হত বুট করার সময় বা এমএস-ডস কোনো অ্যাপ্লিকেশন চালানোর সময়। বুট করার পরেই পর্দায় ফুটে উঠত উইন্ডোজ-৯৫-এর লোগো, গুই শুরু হয়ে যেত। যদিও, চাইলে, এমএসডস-ডট-সিস (msdos.sys) ফাইলকে বদলিয়ে (Bootgui = 0 থেকে Bootgui = 1) ডস-এর কমান্ড প্রম্পট কনসোলেই শুরু করা যেত।

১৯৯৮-এ মাইক্রোসফট নিয়ে এল উইন্ডোজ নাইন্টিএইট। যা ৯৫-এর ওএস কাঠামোটাকে কিছুটা বদলে দিল। কিন্তু ৯৫ বা ৯৮ দুই উইন্ডোজ-এই ব্যাপকভাবে রয়ে গেল ইনটেল এর ১৬ বিট অ্যাসেম্বলি ল্যাংগুয়েজ প্রোগ্রামিং। এই ৮ বিট ১৬ বিট এগুলো নিয়ে আমাদের আলোচনাটা, শূন্য আর দুই নম্বর দিন থেকে, মনে করুন। ৮ বিট এর চেয়ে ১৬ বিট উন্নততর এবং জটিলতর ব্যবস্থা — কম্পিউটার যখন তার তথ্য পড়া বা লেখা বা মনে রাখার একক হিশেবে ব্যবহার করে ১৬ বিট। যে সময় ১৬ বিট ব্যবস্থা আভ্যন্তরীন ভাবে নিয়ে উইনডোজ ৯৮ বাজারে এল, তখন সময় দাবি করছে উন্নততর ৩২ বিট ব্যবস্থাকে। প্রচুর অ্যাপ্লিকেশন সফটওয়ার এবং ওপন সোর্স জগত তখন প্রায় গোটাটাই ৩২ বিট গঠনে চলে গেছে। এখন, এই ২০০৩ সালে যেমন সময়ের দাবি ৬৪ বিট আর্কিটেকচার। ইতিমধ্যেই একাধিক সার্ভার-মাইক্রোপ্রসেসর এবং পিসি-মাইক্রোপ্রসেসর বাজারে এসে গেছে যারা কাজ করে ৬৪ বিটে। তাদের ওএস এবং অ্যাপ্লিকেশন সফটওয়ারও ইতিমধ্যেই আসতে শুরু করেছে ওপন-সোর্স জগতে।

এর পরবর্তী আর একটা মাইক্রোসফট ওএস হল উইনডোজ এনটি। এনটি এখানে নিউ টেকনলজির সংক্ষেপ। যা এক অর্থে উইনডোজ ৯৫-এর সঙ্গে কম্পেটিবল, অর্থাৎ ৯৫-এর সব প্রোগ্রামই এনটি-তে চালানো যাবে, কিন্তু এনটি হল উইনডোজ জগতে প্রথম পূর্ণাঙ্গ ৩২ বিট ব্যবস্থা। মাইক্রোসফট আশা করেছিল যে এনটি বাজারে এসে পুরোনো সমস্ত এমএস-ডস এবং এমএস-উইন্ডোজ ওএস-কে অপসারিত করে দেবে। কিন্তু সেটা ঘটেনি। মূলত এনটির অত্যধিক পোকাধরা বা বাগড সফটওয়ারের জন্যে। তখন মাইক্রোসফট বাজারে আনে ৯৮-এর একটা নতুন সংস্করণ — উইনডোজ এমই, বা মিলেনিয়াম এডিশন। এটা মাইক্রোসফটের বানিজ্যিক নীতিরই অংশ যে একটা অপারেটিং সিস্টেমের পোকা এবং খুঁত গুলোকে বেছে ঠিক করার জায়গায় আর একটা নতুন নামে আর একটা নতুন অপারেটিং সিস্টেম নিয়ে এসো। এর পোকাগুলো ধরা পড়তে পড়তে আবার নতুন একটা। সেই ভাবে বাজারে এল উইন্ডোজ এক্সপি। উইন্ডোজ এনটিতে কেন্দ্রীয় ঝোঁকটা ছিল সার্ভার মেশিন বা নেট-ওয়ার্ক। আর ৯৫, ৯৮ বা এমই-তে মূল ঝোঁকটা ছিল পিসি। মাইক্রোসফটের কথা অনুযায়ী, উইনডোজ এক্সপি-তে এসে এই দুটো ধারা ফের মিশে গেল।

ওয়ার্কস্টেশন বা উচ্চক্ষমতাসম্পন্ন নেটওয়ার্ক সার্ভার মেশিনগুলোয় ইউনিক্স বা লিনাক্স উইনডোজের চেয়ে অনেক বেশি ব্যবহার হয়। বিশেষত যেসব মেশিনে উচ্চ-শক্তির আরআইএসসি (RISC — **R**educed-**I**nstruction-**S**et-**C**hip) ব্যবহার হয়, সেখানে ইউনিক্স বা গ্নু-লিনাক্সই প্রায় একচ্ছত্র। আরআইএস চিপ হল মাইক্রোপ্রসেসরের এমন এক ডিজাইন যেখানে কম্পিউটেশন প্রক্রিয়াটা অনেক দ্রুত এবং দক্ষ রকমে হয়, একটা খুব ছোট আদেশ-মালা ব্যবহার করে। স্বাভাবিকের চেয়ে ছোট এই আদেশমালাটা তৈরি করা হয় এমনভাবে যাতে এদের ব্যবহার করে অপারেটিং সিস্টেম তথা সফটওয়ার কম্পিউটারের প্রসেসরকে পরবর্তী জটিলতর আদেশগুলো পৌঁছে দিতে পারে। এই ছোট আদেশমালার প্রতিটি আদেশ সেই অর্থে অপটিমাইজড মানে সর্বোচ্চ ক্ষমতায় কাজ করার জন্যে নির্মিত এবং নির্দ্ধারিত, এবং সচরাচর এরা কাজ করে কম্পিউটারের সময়একক এক ক্লক সাইকেলের মধ্যে। আরআইএস চিপরা সরল সোজাসাপ্টা আদেশগুলো সিআইএস (CIS — **C**omplex-**I**nstruction-**S**et) চিপের চেয়ে অনেক বেশি গতিতে করতে পারে। এই সিআইএস চিপ বানানো হয় যাতে তাদের বিভিন্ন রকম কাজে ব্যবহার করা যায় — জেনেরাল পারপাজ। এরা অনেক জটিলতর আদেশও নিতে পারে, কিন্তু তাদের গতি আরআইএস চিপের চেয়ে অনেক কম।

পেন্টিয়াম, অ্যাথলন বা এই গোত্রের অন্যান্য স্বাভাবিক সাধারণ সিপিইউ-র মেশিনের দুনিয়ায়, পিসি-ব্যবহারকারীদের মধ্যে, উইনডোজের সবচেয়ে বড় প্রতিদ্বন্দ্বী এখন গ্নু-লিনাক্স। ২০০৩-এর অগাস্টে, মাইক্রোসফটের এক ঘোষণায় গেটস তাদের সবচেয়ে বড় প্রতিবন্ধক হিশেবে বিশ্ব জুড়ে সামগ্রিক ব্যবসায়িক মন্দার পরই এনেছেন ওপন-সোর্স সফটওয়ারকে। ক্রমশ বিশ্ববিদ্যালয়ের অ্যাকাডেমিক কম্পিউটার ব্যবহারকারীদের গণ্ডি ছাড়িয়ে গ্নু-লিনাক্স ব্যবহার করতে শুরু করেছে বিভিন্ন ব্যবসায়িক প্রতিষ্ঠানও। গ্নু-লিনাক্স ক্রমে অনেক ইউজার ফ্রেন্ডলি-ও হয়ে উঠেছে। বিশেষত গুহনোম কেডিই ইত্যাদি বিভিন্ন রকম গুই-এ কমান্ড প্রম্পট বাদ দিয়ে মাউসে কাজ করার সুযোগ আসায়, এবং মাল্টিমিডিয়া আর গ্রাফিক্সে উইনডোজের তুলনায় যোজন যোজন উৎকৃষ্ট সফটওয়ার তৈরি করায়।

ইউনিক্স এবং গ্নু-লিনাক্সে গুই এসেছিল এক্স-উইনডোজ (X-Windows) নামে, যাকে তৈরি করে এমআইটি, ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনলজি। যেখানে কমান্ড প্রম্পটের কাজগুলো ছবিতে মাউস টিপে করা যায়। আমাদের এই পাঠমালায় যার খুঁটিনাটিগুলো আমরা আনবনা, আগেই বলেছি। এই পাঠমালায় গোটাটাই আমরা করব কমান্ড প্রম্পটে।

আর উইনডোজের তুলনায় অনেক বেশি সমর্থ বা রোবাস্ট গ্নু-লিনাক্স চিরকালই ছিল। রিস্টার্ট বা হ্যাং-এর দুস্বপ্ন নিয়ে রাতে ঘুমোতে যায়না এমন উইনডোজ ব্যবহারকারী অসম্ভব, আর গ্নু-লিনাক্সে এরা প্রায় অজানা শব্দ। ইন্টারনেট যেহেতু প্রাথমিক ভাবে তৈরি হয়েছিল ইউনিক্স বা গ্নু-লিনাক্সের হাতে এবং জন্যে, স্বাভাবিকভাবেই ইন্টারনেট ব্যবহার করাও উইনডোজের তুলনায় ইউনিক্সে ও লিনাক্সে অনেক সহজ ও সুঠাম। এবং গত কয়েক বছর যেকোনো কম্পিউটার ব্যবহারকারীর কাছে একটা জ্যান্ত ত্রাস হল কম্পিউটার ভাইরাস। কত তথ্য রসদ এবং অর্থ যে বলি হয়েছে এই ভাইরাসে তার কোনো ইয়ত্তা নেই। পরে, গ্নু-লিনাক্স কাঠামোটা আমরা যখন বুঝতে শুরু করব, বিশেষত রুট বা সুপারভাইজার এবং ইউজার বা ব্যবহারকারীর বিভিন্ন অধিকারের পার্থক্যগুলো, তখন বুঝব যে, গ্নু-লিনাক্স এমন করেই তৈরি যাতে ভাইরাসের পক্ষে এই ধরনের কাঠামোয় দাঁত ফোটানো সম্ভব না। আর ইউনিক্স-এর বেলাতেও এগুলো সত্যি, কিন্তু বিভিন্ন কোম্পানির ব্রান্ডনেম প্রোডাক্ট ক্লোজড সোর্স ইউনিক্সগুলো এত দামি যে কোনো সাধারণ ব্যবহারকারীর কাছে, শুধু আমাদের দেশে নয়, পশ্চিমেও, কেনা প্রায় অসম্ভব।

এই চতুর্থ জেনারেশন কম্পিউটারে অপারেটিং সিস্টেমের স্তরে একটা নতুন জিনিষ এসেছিল আশির দশকের মাঝামাঝিতে এসে। যার নাম নেটওয়ার্ক-ওএস বা ডিস্ট্রিবিউটেড-ওএস। এর মূল কারণটা এই যে ক্রমশ আরো বেশি বেশি সংখ্যক ব্যক্তিগত পিসি এবং সার্ভার বা ওয়ার্কস্টেশন পরস্পর সংযুক্ত হয়ে উঠছিল ল্যান ইত্যাদি দ্বারা, মূলত ব্যবসার এবং তথ্য চলাচলের প্রয়োজনে। এই ধরনের ওএস-এ একজন ব্যবহারকারী তার নিজের কম্পিউটার ছাড়া অন্য কম্পিউটারগুলোও ব্যবহার করার সুযোগ পায়, সেখান থেকে কোনো ফাইল কপি করে নিতে পারে নিজের মেশিনে, নিজের মেশিন থেকে ফাইল পাঠাতে পারে অন্য মেশিনে, এবং এই জাতীয় আরো অনেক কিছুর সুযোগ থাকে। অর্থাৎ, গ্লোবাল স্তরে মানে সবগুলো মেশিন মিলিয়ে থাকে একটা নেটওয়ার্ক এবং তার নেটওয়ার্ক ওএস। আর লোকাল স্তরে একটা লোকাল মেশিনে থাকে এক বা একাধিক লোকাল ব্যবহারকারী এবং তাদের একটা লোকাল ওএস। এইখান থেকে সততই পরের পদক্ষেপটা আসে খুব সহজে — ঠিক ওই তৃতীয় জেনারেশনের টাইম-শেয়ারিং-এর মূল ধারণাটা মনে করুন। স্ট্যাটিস্টিক্সের সেই সোনালি নিয়ম অনুযায়ীই, যে কোনো মুহূর্তেই, অনেকগুলো মেশিনের অনেকগুলো প্রসেসর জাস্ট ঝিমোচ্ছে, কারণ তারা তো তাদের ব্যবহারকারীদের মত কফি খেতে বা আড্ডা মারতে পারেনা। তাহলে কেন সেই ঝিমোতে থাকা প্রসেসরদের ঝুলে থাকা সমগ্র শক্তিকে কাজে নিয়ে আসা যাবেনা? যেখানে সব প্রসেসরদের মিলিত ক্ষমতাটা সিস্টেমের প্রসেসিং বা কম্পিউটিং পাওয়ার হয়ে উঠবে? ডিস্ট্রিবিউটেড ওএস যেখানে তার ওই একগুচ্ছ প্রসেসরকে মোট রসদের একটা ভাণ্ডার হিশেবে দেখবে, এবং যখনই যে প্রসেসর তার ক্ষমতার চেয়ে কমে কাজ করছে তাকে নতুন কাজ দেবে। যেভাবে সে অন্য রসদগুলো নিয়ে শেয়ারিং এবং মাল্টিপ্রোগ্রামিং করত। এইভাবেই এল নেটওয়ার্ক ওএস বা ডিস্ট্রিবিউটেড ওএস।

একটা একক লোকাল অপারেটিং সিস্টেমের সঙ্গে ডিস্ট্রিবিউটেড বা নেটওয়ার্ক অপারেটিং সিস্টেমের কিন্তু কোনো মূলগত পার্থক্য নেই। এখানেও একাধিক মেশিনের মধ্যে সংযোগ ঘটতেই পারে। অবশ্যই একটা ল্যান কার্ড বা মোডেম জাতীয় কোনো সংযোগের যন্ত্র থাকবে, এবং সেই যন্ত্র চালানোর জন্যে আর যন্ত্রে আদান-প্রদান করা তথ্যের প্রবাহ নিয়ন্ত্রণের জন্যে থাকবে একটা ড্রাইভার, এবং থাকবে কিছু সফটওয়ার যা দিয়ে দূরবর্তী মেশিনগুলোর সঙ্গে সংযোগ এবং তথ্য দেওয়া-নেওয়া করা যায়। কিন্তু এতে ওএস-এর মৌলিক গঠনের কোনো বদল ঘটেনা। অন্যদিকে, একটা ডিস্ট্রিবিউটেড ওএস-এর একটা মেশিনে বসে ব্যবহার-করার সময় একজন ব্যবহারকারীর কাছে কিন্তু সেটা একটা অভ্যস্ত লোকাল ওএস বলেই মনে হবে। আলাদা আলাদা মেশিনের আলাদা আলাদা হার্ড ড্রাইভ যেন আলাদা আলাদা ডিরেক্টরিতে তথ্যের আলাদা আলাদা ভান্ডার। যদিও এখানে মোট কম্পিউটেশন প্রক্রিয়াটা চলছে অনেকগুলো প্রসেসরকে মিলিয়ে। একজন ব্যক্তি ব্যবহারকারীর জানার বা বোঝার প্রয়োজনও নেই যে আসলে ওএস কোন প্রোগ্রামকে কোন মেশিনের কোন ডিরেক্টরি থেকে তার জন্যে চালাচ্ছে। বা কোন কোন সিস্টেম ফাইল কোন কোন মেশিনের কোথায় রাখা। এই গোটা কাজটাই অভ্যস্ত দক্ষ নিঃশব্দ রকমে করে চলবে ওএস, অবসরহীন দাস।

১৩।। একই সময়ের শরীরে বিভিন্ন সময়

একটা মজার জিনিষ এই যে কম্পিউটারের তথা অপারেটিং সিস্টেমের ইতিহাসে প্রায় একটা উপাদানও কিন্তু আজো অবশেষহীন ভাবে অবলুপ্ত হয়নি। এক একটা নতুন উপাদান পদক্ষেপ আবিষ্কার ঘটেছে এবং তাদের ব্যবহার করা শুরু হয়েছে। আবার এসেছে একটা নতুন কিছু। কিন্তু, তাই বলে পুরোনো উপাদানটা নিশ্চিহ্ন হয়ে যায়নি। এই ২০০৩ সালের মধ্যেই, চতুর্থ জেনারেশন মেশিনের মধ্যেই, প্রায়ই বেশ উজ্জ্বল রেলায়, বেঁচে আছে একটু ’৬২, একটু ’৭৪, একটু ’৮৫ বা একটু ’৯৩। ধরুন মেইনফ্রেম বা মিনিকম্পিউটার বা পার্সোনাল কম্পিউটার বা এমবেডেড সিস্টেম — এরা আলাদা আলাদা ভাবে গজিয়ে উঠেছে ইতিহাসের বিভিন্ন মুহূর্তে এবং বিভিন্ন ধরনের জ্যান্ত প্রয়োজনকে মেটাতে। কিন্তু এরা একইসঙ্গে সবাই কাজ করে চলেছে। মোট সভ্যতায়, মোট বাজারে, মোট কম্পিউটার বিজ্ঞানে।

প্রথমদিকের মেইনফ্রেম মেশিনগুলোয় প্রোগ্রাম বানানো হত অ্যাসেম্বলি লেভেল ভাষায়। এমনকি কম্পাইলার বা অপারেটিং সিস্টেম গোছের ব্যাপক জটিল এবং প্যাঁচালো প্রোগ্রামগুলোও লেখা হত অ্যাসেম্বলিতে। মিনি-কম্পিউটার যখন সিনে এল তখন মেইনফ্রেম মেশিনে বহুল ব্যবহার হচ্ছে ফোর্ট্রান বা কোবোল (COBOL — **Co**mmon-**B**usiness-**O**riented-**L**anguage — প্রায় স্বাভাবিক ইংরিজি, ১৯৫৯ থেকে ৬১-র মধ্যে তৈরি হয়, ব্যবসার রোজকার প্রয়োজন মেটাতে) ইত্যাদি হাই-লেভেল ভাষা, মানে মানুষের ভাষার খুবই কাছাকাছি যারা। কিন্তু তখনো, তখন নতুন আসা মিনি-কম্পিউটারগুলোয় প্রোগ্রাম করা হত ওই অ্যাসেম্বলি কোডেই, মূলত মেমরি বা স্মৃতির অভাবের কারণেই। তারপর যখন মাইক্রো-কম্পিউটার মানে প্রথম যুগের পিসি বাজারে এল, তাদেরও প্রোগ্রাম করা হত অ্যাসেম্বলি কোডে। যদিও, মজার কথাটা এই, ততদিনে মিনি-কম্পিউটারে প্রোগ্রাম করা হচ্ছে হাই লেভেল ভাষায়। তার অনেক পরে, নিকট অতীতে, যখন পাম-টপ কম্পিউটার এল, তাদেরও শুরুর দিকে প্রোগ্রাম করা হত অ্যাসেম্বলিতে। স্মার্ট কার্ড ইত্যাদি এমবেডেড সিস্টেম-এর বেলাতেও তাই।

ওএস-এর ক্ষেত্রে, একদম গোড়ার দিকে, মাল্টিপ্রোগ্রামিং-এর জন্য প্রয়োজনীয় প্রোটেক্টিভ হার্ডওয়ার, মানে প্রত্যেকটা রসদ এবং প্রোগ্রামকে তার নিজের নিজের মহলে অন্য প্রোগ্রামের প্রকোপ থেকে বাঁচাতে পারে এমন হার্ডওয়ার, ছিলনা। যেমন আমরা আগেই বলেছি। তাই, তখনকার ওএস একসঙ্গে একটার বেশি প্রোগ্রাম চালাতে পারত-না। মাল্টিপ্রোগ্রামিং তখন সম্ভব ছিলনা। প্রোগ্রামগুলো আবার মেশিনে ঢোকাতে হত হাতে করে। পরে যখন প্রয়োজনীয় হার্ডওয়ার এল, একসঙ্গে একাধিক প্রোগ্রাম চালাতে এবং নিয়ন্ত্রণ করতে শিখল ওএস। তারপর, সময়ের হাত ধরে, নতুনতর হার্ডওয়ারের হাত ধরে, একসময় এল পুরোপুরি পূর্ণমাত্রার টাইমশেয়ারিং। মিনি-কম্পিউটারের প্রথম দিকে, তাদেরও কোনো প্রতিরক্ষামূলক হার্ডওয়ার ছিলনা, তাই সেখানেও মাল্টিপ্রোগ্রামিং করা যেতনা। যদিও, মেইনফ্রেম মেশিনের পৃথিবীতে মাল্টিপ্রোগ্রামিং ততদিনে জলভাত হয়ে গেছে। ক্রমে মিনি-কম্পিউটারেও এল প্রয়োজনীয় হার্ডওয়ার, তার মানে সেই পথে একদিন এল মাল্টিপ্রোগ্রামিং, একই পথে একদিন টাইম-শেয়ারিং-ও এল। পামটপ কম্পিউটার বা স্মার্ট কার্ডের বেলাতেও তাই। অভিযোজনের যে সিঁড়ি বেয়ে একসময় মানুষ নামের প্রজাতিটা এসেছে, প্রতিটি মানুষের ভ্রূণকে তার মায়ের গর্ভে ঠিক যেমন সেই প্রতিটি সিঁড়ি পার হয়ে হয়ে আসতে হয়। তার শরীরে ফুলকার চিহ্ন আসে, আবার মিলিয়ে যায়, সরীসৃপের বুকে হাঁটা অস্তিত্বের পদচিহ্ন আসে, আবার মিলিয়ে যায়, একদিন আবার তারও সন্তান হয়, সে-ও আসে সেই একই পথ পেরিয়ে। বারমুডা আর পুয়ের্তোরিকোর ভিতর সারগাসো সমুদ্রে যেমন ডিম ফুটে বেরোয় ঈল মাছের ছানা, লেপ্টোসেফালি থেকে ঈল হয়, যায় নদীতে, সারাজীবন কাটায় নদীর মিষ্টিজলে, আবার মৃত্যুর আগে ফিরে আসে সারগাসো সমুদ্রের নোনা জলে, ডিম পাড়ে, সেই একই বৃত্ত চলতে থাকে।

হার্ডডিস্ক প্রথম লাগানো হয়েছিল মেইনফ্রেমে, তারপর তা এল মিনিতে, পরে মাইক্রোতে। প্রথম যখন হার্ডডিস্ক এল, তার হাত ধরে এল আদিম ধরনের ফাইলসিস্টেম। ১৯৬০-এর দশক জুড়ে পৃথিবীর সবচেয়ে শক্তিশালী মেইনফ্রেম কম্পিউটার সিডিসি ৬৬০০-তে, একজন ব্যবহারকারী সবচেয়ে বেশি যা করতে পারত তা হল, একটা হার্ডডিস্কে একটা ফাইল তৈরি করা এবং তারপর সেটাকে স্থায়ী বলে ঘোষণা করা, নইলে তার নিজের প্রোগ্রাম থেকে বেরনো মাত্র সেই ফাইল হার্ড ডিস্ক থেকে মুছে যাবে। ওই ফাইল পরে আবার ব্যবহার করার পরিকল্পনা থাকলে প্রোগ্রামের সঙ্গে একটা বিশেষ কমান্ড ব্যবহার করে ফাইলের সঙ্গে একটা পাসওয়ার্ড লাগাতে হত। কারণ, সব যাবতীয় সমস্ত

ব্যবহারকারীর জন্যে ডিরেক্টরি ছিল মাত্র একটা। প্রতিটি ইউজারের প্রতিটি ফাইলই থাকত সেই একমেবাদ্বিতীয়ম ডিরেক্টরিতে। ইউজারদেরই মাথায় রাখতে হত যাতে একজনের দেওয়া একটা ফাইল-নামের সঙ্গে অন্যের দেওয়া অন্য কোনো ফাইলের নাম মিলে না-যায়। পরে যখন আমরা গ্নু-লিনাক্স ফাইলসিস্টেম খুব বিশদে আলোচনা করব সাত আর আট নম্বর দিনে, তখন দেখবেন ফাইলব্যবস্থার সেই প্রারম্ভিক কাঠামোগুলো কী রকম ফিরে ফিরে এসেছে।

ভারচুয়াল মেমরি বলতে আমরা বুঝি মেমরির ভৌত পরিমাণের চেয়ে বেশি স্মৃতিকে ব্যবহার করতে পারা। আমরা উল্লেখ করেছি আগেই, সোয়াপ ফাইলের কথা বলেছি। সোয়াপ-ফাইল ঠিক এইটাই করে, ফিজিকাল বা ভৌত স্মৃতিকে নিজের বাইরে বাড়িয়ে তোলে ভারচুয়াল বা ভৌতিক স্মৃতি বানিয়ে। ধরুন আপনার মেশিনে ২৫৬ এমবি র‍্যাম আছে, মানে স্মৃতির ভৌত পরিমাণটা হল ২৫৬ এমবি মানে ২৫৬ × ১০২৪ × ১০২৪ × ৮ = ২,১৪৭,৪৮৩,৬৪৮ সংখ্যক বিট বা খোপ — যার প্রতিটি খোপে আসতে পারে হয় একটা ১ অথবা একটা ০। এবার, সোয়াপ ফাইল কাকে বলে মনে করুন, হার্ড ডিস্কে অস্থায়ী ভাবে কিছু লিখে রাখার একটা জায়গা। এবার ধরুন আপনার মেশিনে সোয়াপ ফাইলের সাইজ ৫১২ এমবি। কোনো প্রোগ্রাম চালানোর সময় প্রোগ্রামের সাইজ, প্রোগ্রামের জন্য ব্যবহৃত মোট তথ্যের সাইজ সব মিলিয়ে সীমাটা নির্দিষ্ট ছিল ২৫৬ এমবি। এখন সোয়াপ ফাইলের দৌলতে অন্তত তার দ্বিগুণ আয়তনের মোট প্রোগ্রাম ও তথ্য আপনি ব্যবহার করতে পারবেন। অর্থাৎ আপনার মেশিনের ফিজিকাল মেমরি ২৫৬ এমবি-কে আপনি অতিক্রম করে যেতে পারলেন ভারচুয়াল মেমরি বা ভৌতিক স্মৃতির দৌলতে। হার্ড ডিস্কে অস্থায়ী ভাবে লেখা ওই সোয়াপ ফাইল তখন আর এক রকম র‍্যামের কাজ করছে। ঠিক কী ভাবে এই বাড়িয়ে তোলাটা ঘটে, সেটা আমরা পরে বুঝব।

এই প্রসঙ্গে একটু বলে নেওয়া ভাল — পরে আমরা আরো বিশদে বুঝব, আমাদের মেশিনে স্মৃতি বা মেমরি আসলে পাঁচ রকমের। এবং প্রোগ্রাম থেকে প্রোগ্রামে মেশিনের গতির যে বিস্তর ফারাক ঘটে তার একটা বড় কারণ এই পাঁচ ধরনের স্মৃতির নিজেদের ভিতরকার গতি এবং আকারের তফাত। এই পাঁচ ধরনের স্মৃতি হল :

(১) মাইক্রোপ্রসেসরের নিজের ভিতরকার রেজিস্টার

(২) আভ্যন্তরীন বা ইন্টারনাল ক্যাশে, মানে, চিপের শরীরেই থাকা স্মৃতি-প্যাকেট

(৩) চিপের বাইরে স্মৃতি-প্যাকেট বা এক্সটারনাল ক্যাশে

(৪) মূল মেমরি মানে যাকে সচরাচর আমরা র‍্যাম বলে ডেকে থাকি

(৫) হার্ড ডিস্ক

ইন্টার্নাল এবং এক্সটারনাল এই দুই ধরনের ক্যাশে (cache) বা স্মৃতি-প্যাকেট হল ছোট ছোট মেমরির প্যাকেটের মত যেখানে কোনো তথ্য বা প্রোগ্রামের পুরোটা বা অংশ মেশিন রাখে যাতে তাদের পাওয়ার সময় হার্ড-ডিস্ক থেকে পড়ার সময়টুকুও ব্যয় করতে না-হয়। ক্যাশে মেশিনের কাজের গতিকে বাড়িয়ে তোলে, কিন্তু একই সঙ্গে, এখনই যা দেখব আমরা, বাড়িয়ে তোলে খরচের পরিমাণও। এবার ইন্টারনাল ক্যাশে হল তাই যার আবাস চিপের ভিতরেই, আর এক্সটার্নাল ক্যাশে মানে তার আবাস চিপের বাইরে। এখানে একটা কথা বলে রাখা ভাল — মেমরির তুলনায় সিপিইউ অনেক বেশি দ্রুত বলেই কিন্তু ক্যাশে ব্যবহার করতে হয়। গল্পটা অনেকটা ভুতের হাতের লেজের মত, নিদ্রা চটিয়ে তাকে তোলার পর, কাজ দেওয়াটাই হয়ে দাঁড়িয়েছিল সমস্যা, শেষে একটা কুকুরের বাঁকানো লেজ তাকে সোজা করতে দেওয়া হয়েছিল। এখানে আবার সেটাও করার জো নেই, খুবই কস্টলি ভুত। তাই, সেই দ্রুতগতিতে কাজ করে চলা রাজমিস্ত্রির সঙ্গে সাধারণের চেয়ে একটু দ্রুততর যোগাড়ে লাগানো হয়, যার নাম ক্যাশে। এরা এমনি যোগাড়েদের পাশাপাশি কাজ করে এবং যোগানোর কাজটাকেই মোটের উপর দ্রুততর করে তোলে। তুলনামূলক ভাবে শ্লথ মেমরিকে দ্রুততর করে তোলা হয়, সিপিইউ-র কাজ করে চলার স্বার্থে তথ্যরাজি স্থানান্তরের যে মোট কাজ থাকে মেশিনে, সেই কাজটাই করে ক্যাশে, আর একটু বেশি গতিতে। মেমরির সঙ্গে প্যালা দেয়। টেকনোলজি তো শুধুই বদলাচ্ছে। কোনোদিন যদি দেখা যায়, মেমরি সিপিইউ-র চেয়ে দ্রুততর হয়ে গেছে, তখন আর ক্যাশে ব্যবহারের কোনো প্রয়োজন পড়বে না।

(১) থেকে (৫) — উপরের এই তালিকাটা করা হয়েছে সবচেয়ে দ্রুতগতি স্মৃতি থেকে সবচেয়ে শ্লথগতি স্মৃতির দিকে। চিপের নিজস্ব রেজিস্টার সবচেয়ে তাড়াতাড়ি কাজ করে — সেকেন্ডে মোটামুটি দশহাজার কোটি বার মেশিন একে পড়তে পারে, এক নম্বর দিনে আমরা এই রেজিস্টারগুলো নিয়ে কথা বলেছি, মনে পড়ছে? আর সবচেয়ে শ্লথগতি হল হার্ডডিস্ক — সেকেন্ডে মোটামুটি একশোবার মেশিন একে পড়তে পারে। তাও, এই হিশেবটা পুরোনো, খুব আধুনিক মেশিনে গতিটা এর চেয়েও অনেকটাই বেশি। কিন্তু গতি বাড়ানো মানেই খরচ বাড়িয়ে তোলা। যে কোনো একটা সময়ে কোনো একধরনের কম্পিউটারে কোন ধরনের স্মৃতি কতটা থাকবে সেটা ঠিক হয় সেই মুহূর্তের বিভিন্ন প্রকৌশলের দাম আর ওই মেশিনে যে কাজ করা হবে তার প্রকৃতি ও পরিমাণের উপর দাঁড়িয়ে। আমি আমার মেশিনকে ব্যবহার করি মূলত লেখার কাজে, আর একটু একটু প্রোগ্রাম শিখতে গিয়ে লেখার কাজে, সেটাকেও প্রোগ্রামিং না বলে লেখা বলাই ভালো। আর গান শোনার বা ফিল্ম দেখার কাজেও ব্যবহার করি, কখনো ছবি আঁকতেও, এই বইয়ের ছবিগুলো যেমন। তাই, আমার যে মেশিনে চলে যায়, থ্রিডি অ্যানিমেশনের কাজ করে এমন কারুর তো তাতে চলবেনা।

যাইহোক, যে কথা আমরা বলছিলাম, অপারেটিং সিস্টেমের বিবর্তনের বিষয়ে — সেটা এই ভারচুয়াল মেমরির ক্ষেত্রেও সত্যি। স্মৃতির ভৌত পরিমাণের চেয়ে বড় আকারের স্মৃতির প্রয়োজন মেটাতে ভারচুয়াল বা ভৌতিক স্মৃতি প্রথম এসেছিল মেইনফ্রেম কম্পিউটারে। তারপর এল মিনিতে। তারপর মাইক্রোয়। ক্রমে ছোট থেকে ছোটতর মেশিনে। নেটওয়ার্কিং-এর ইতিহাসও এই একই।

তিন নম্বর দিনের ভূমিকায় যা আমরা বলেছিলাম, সফটওয়ার কী পথে বিকশিত হবে, সেটা নির্ভর করে সেই যুগের মানবসমাজের কাজের প্রয়োজনের উপর। আর সফটওয়ারের বাস্তব বিকাশ সবসময়ই নির্ভর করে থাকে সেই যুগের হার্ডওয়ারের উপর। প্রথম দিকের মাইক্রো-য় মেমরি ছিল মাত্র ৪ কেবি, আর কোনো প্রতিরক্ষামূলক হার্ডওয়ার তাতে ছিলনা। ওই রকম কোনো মেশিনে হাইলেভেল ভাষা বা মাল্টিপ্রোগ্রামিং ব্যবহার করা অসম্ভব। তারপর মাইক্রোরা বদলাতে বদলাতে আজকের পিসি হয়ে উঠল, তাদের হার্ডওয়ার বদলে গেল, তখন বদলাল তাদের সফটওয়ার সমাহারটাও। আজ তিরিশ বছর আগের মেইনফ্রেমের গল্প শুনে অবাক হই, কিন্তু, এই বদল আরো বাড়বে। আরো তিরিশ বছর বাদে অন্য কেউ যদি পড়ত এই লেখাটা, পড়বেনা, কারণ বিংশ শতাব্দীতে মানুষের শোকের আয়ু একবছর, ওয়েবপেজ বড়জোর দুতিনশো মিনিট, তার কাছে আমাদের এই অবাক হওয়াটা খুব অবাক লাগত। হাসত সে। মনে হত, ওই পৌরাণিক সিস্টেম নিয়ে এরা এত উচ্ছ্বাস করেছে? যেমন আমরা সিঁদুরচরণের কথা পড়ে হাসি, বনগাঁর গ্রাম থেকে নদীয়া অব্দি গিয়ে হতভম্ব হয়ে যে ভেবেছিল, 'এই পিথিমির কি কোনো সিমেমুড়ো নেই?'

ইউনিক্স-এর দুটো বই, দুটোরই চ্যাপ্টার এক, আমাদের খুব কাজে আসবে। একটা সুমিতাভ দাস-এর, ইউনিক্স - দি আল্টিমেট গাইড। আর অন্যটা কারনিঘান অ্যান্ড পাইক-এর ইউনিক্স - দি প্রোগ্রামিং এনভায়রনমেন্ট। আর, ট্যানেনবম-এর মডার্ন অপারেটিং সিস্টেম তো আছেই। যদি কেউ এই স্মৃতির ব্যাপারটা বেশ বিশদে পড়তে চায় তার জন্যে জিএলটির সিডির 'টোটাল হাউ-টু ফ্রম টিএলডিপি' থেকে 'ইউনিক্স অ্যান্ড ইন্টারনেট হাউ-টু'-টা বেশ ভাল টেক্সট। এর পরের দিন আমরা সরাসরি ঢুকব গ্নু-লিনাক্স একটা সিস্টেমে, তার বুট প্রক্রিয়া থেকে শুরু করে।